Contents

Dytran Release Guide

1 Introduction

2 New Capabilities in HPC and Fluid-Structure Interaction (FSI)
   DMP for FSI Applications with Adaptive Euler  8
   DMP Improvements for Coupling and Editing  12
   DMP Support for Importing Eulerian Archive Files for Subsequent Simulations  14
   Other Enhancements and Defect Corrections  17

3 System Information
   Software Installation  24
   Licensing  24
   Release Platforms  25
   Memory Requirements  26

4 Using Dytran
   Running Dytran on Unix and Linux  28
   Running Dytran on Windows  28
   Postprocessing Dytran Results  29
   Postprocessing Dytran Results of Windows on UNIX  29
   Postprocessing Dytran Results with ParaView  29
   Time History Plots  45
1 Introduction
Dytran™ 2012 is the latest and most comprehensive version of Dytran released by MSC.Software, bringing new simulation technology and improved performance.

Dytran 2012 is available on UNIX (HP-UX PA- RISC 2.0, HP-UX Itanium2), IBM RS/6000 (Power 5), Sun SPARC Solaris, Solaris X64, Linux Itanium2 (Red Hat 4, Update 5), Linux Itanium2 SGI Altix (SGI ProPack5), Linux X8664 (Red Hat 5, Update 4), Linux 32 (Red Hat 5, Update 4), Windows 32 bit (Windows7) and Windows 64 bit (Windows7) platforms. Please see System Information in Chapter 3 of this guide for more details.

Dytran 2012 includes major new capabilities that are primarily focused in the areas of High Performance Computing (HPC) for fluid-structure interaction (FSI) applications resulting in dramatic performance improvement for CPU intensive simulations. These are:

- Distributed Memory Parallel (DMP) support for FSI applications with Adaptive Meshing and load balancing.
- DMP support for importing Euler Archive files for subsequent situations
- DMP Improvements for Coupling and output generation
- Enhancements to \texttt{FFCONTR} to initialize air pressure inside the fluid filled bottles
- \texttt{MESH, BOX} is enhanced to create Eulerian meshes in a user-defined local coordinate system. In the previous releases, the creation of Eulerian meshes was limited to global coordinate system using the \texttt{MESH, BOX} entry.
- Output requests were extended to include “ALL” specifiers to generate time history output for all materials, all Lagarangian material, and all Eulerian material.

In addition, several critical software defects have been corrected in this release.

The Dytran 2012 DMP technology is not extended to the structural solver. The DMP capability does not require any additional licensing requirements. Please see Licensing in Chapter 3 of this guide for more details.


Dytran uses the Macrovision FLEXlm™ licensing system. If you already have a Dytran 2010 license, you will not need to obtain a new authorization code to activate Dytran 2012 on your computer. However, you will need to install the latest FLEXlm 11.9 license server.

If you need assistance while installing Dytran 2012, please call the MSC Technical Support Hotline at 1-800-732-7284, or E-mail your support questions to \texttt{mscdytran.support@mscsoftware.com}. 
New Capabilities in HPC and Fluid-Structure Interaction (FSI)

- DMP for FSI Applications with Adaptive Euler 8
- DMP Improvements for Coupling and Editing 12
- DMP Support for Importing Eulerian Archive Files for Subsequent Simulations 14
- Other Enhancements and Defect Corrections 17
DMP for FSI Applications with Adaptive Euler

The DMP in Dytran is extended to include the support for adaptive Euler. The adaptive meshing technology in Dytran automatically generates the Eulerian meshes as needed during the analysis. When the structure undergoes severe deformation, the adaptive mesh is expanded to follow the movement of coupling surfaces, preventing the creation of extraneous elements at the start of the analysis and therefore resulting in significant reduction in simulation runtime. The DMP support for adaptive Euler capability allows the users to run the CPU intensive FSI applications on multiple cores and further improves the performance runtimes. There are no GUI requirements for this new capability.

One important feature in DMP support for adaptive mesh is the load balancing schema. During the DMP simulation with adaptive mesh, the Eulerian mesh is decomposed into many domains and each domain is spawned onto different processors. The load balancing is intended to ensure that all cores have about the same amount of work for an efficient computation as the meshes are transported from one core to the next during the analysis. There are many FSI applications that can result in poor performance due to lack of proper load balancing. For optimal speed-ups, the Euler mesh has to be re-partitioned across the cores. For example, during the fluid-filled bottle droptest on two cores, as the fluid moves through the stationary Euler mesh, it leaves one core and enters the second core. As a result, one core is off-loaded as the second core is loaded. Load balancing ensures that both cores have a balanced amount of processing, and it is also a major step towards good scalability on multiple cores.

Adaptive meshing can result in shorter run times for tank sloshing simulations especially when there is significant movement of the tank through the Euler mesh. Running these simulations with adaptive meshing in combination with DMP may result in poor performance in some cases. This happens because of inefficient load unbalancing across the processors. When the tank moves, the adaptive mesher creates Euler elements at the front of the tank and deletes the elements at the back of the tank. Consequently, elements are deleted on one processor and created on another. This causes load unbalance that might lead to poor performance. To overcome load unbalancing, Euler cubes can be moved from one processor to the other processor when there is sufficient number of cubes present in the simulation. But if the number of Euler cubes is not sufficient, the re-allocation of cubes is not possible and causes load imbalance.

Whether or not load imbalance problems have occurred, it can be checked by the message in the OUT file:

```
********************************************************************
*                                                                  *
* MESH (ID=101) HAS BEEN REDISTRIBUTED ACROSS CPUS AT CYCLE 191400 *
*                                                                  *
* NUMBER OF FINITE VOLUME ELEMENTS ON EACH TASK:                  *
*                                                                  *
* TASK    0                     :   197772                        *
* TASK    1                     :    52013                        *
*                                                                  *
********************************************************************
```

In the example above, there are only two cubes for the entire tank sloshing that were used, and the message shows the load unbalance. By adding more cubes in the direction of tank motion, there will be more cubes to move around and will resolve the load unbalance problems. Caution must be exercised in creating excessive number of cubes since it can lead to overhead costs. For this reason, the number of
cubes should be limited to about 100. The overhead costs for simulations with adaptive meshing on eight or more cores may result in degraded performance.

Example 1 – Fluid Filled Bottle Drop Test

The following example was run with DMP on eight cores with Adaptive Euler:
Performance Table
Example 2 – Airbag

1 CPU

2 CPUs

4 CPUs

8 CPUs
Performance Table

DMP Improvements for Coupling and Editing

Several performance benchmarks have been conducted in previous releases to first identify the bottleneck areas in the code that are CPU intensive, and second, to improve the performance. The coupling and output generation (aka “editing” among users) calculations during a DMP FSI application has been identified as one of the major bottlenecks where it consumes a significant amount of CPU time in Dytran. This is clearly evident as the numbers of the processors are increased (see figure below for 2010).

The new performance enhancement to coupling and output generation algorithms dramatically improves the performance on FSI applications as shown below.
Each color on the curves demonstrates a different benchmark. As shown in the overall performance charts at the top, most of the benchmarks in 2010 release shown in the charts to the left are around 3-5.5 range (performance numbers are compared to one core) whereas the same benchmarks run with 2012 version start at around 5 at a minimum and go up in performance up to 20 as the number of processors are increased to 32 cores. There is one problem that is outlier (shown in light green at the top of same chart) which did not experience any significant improvement.

The greatest performance improvements were achieved in output generation shown in second chart from the top where all benchmarks achieved dramatic speedups. Integration schemes including sub-cycling
without initialization and output generation as well as clumping and coupling areas in the code were also improved in performance in 2012 release.

**DMP Support for Importing Eulerian Archive Files for Subsequent Simulations**

The DMP support for importing Euler ARC dramatically enhances the performance for applications such as blast.

In blast wave simulations, the Euler mesh has to be sufficiently fine enough to capture the initial expansion and propagation of the blast wave through the medium. Once the blast wave has expanded enough to reach its target, the blast wave characteristics are recorded in the Archive output file and the blast wave results are used as an initial state in the subsequent runs where a much coarser mesh is constructed instead. Running the initial full model with both structure and explosives with fine mesh is CPU intensive and requires a much longer simulation time.

This capability is especially useful in blast simulation where the distance between the structure and the blast location is often large, resulting in excessive CPU time for the blast wave to reach the structure. In addition, the simulation may have to be repeated several times for different structures or for different positions. To reduce the simulation time, the simulation is split up in two parts. In the first simulation, the structure is omitted. In subsequent runs, the result can be imported into a simulation which includes the structure. The blast wave almost immediately hits the structure at the start of the run without using a lot of CPU time. In addition, importing the Euler archive also allows mapping the fine mesh results to a coarser mesh resulting in significant performance improvement and accuracy.

Another technique to reduce the simulation time in the blast applications is to take advantage of the symmetry boundary conditions where only a small portion of the entire model such as half-symmetry or quarter symmetry is constructed for the analysis. The limitation of this technique is that not all models are symmetric. Therefore, using an initial fine mesh to capture the blast wave physics followed by a coarse mesh in subsequent runs is a general approach for all blast wave application. This new capability in Dytran 2012 will allow the users to import the Archive files using the DMP to speed up the simulation.

**Blast Wave Example with Single Domain**

This is a typical example where the fine Eulerian mesh is used to capture the details of the blast wave, and the results are stored in ARC file. In subsequent runs, a coarser mesh of Euler is used with a fine structural mesh. The blast wave from initial run is mapped on to the structure in the second run using the DMP capability on four processors to speed up the simulation.

The user has to do the following steps:

1. Create a model with small but fine Euler mesh.
2. Run the model and see when the blast wave approaches the boundary of the Euler mesh. Select a time for this.
3. Create a coarse and larger 3-D model and use the option `EULINIT/eid` to point to the archive of the fine but small 3-D model. Also, add the selected time to `EULINIT/eid`.

4. If required repeat the above steps a few times

This example uses eight cores in DMP mode.
Initial Remapped Pressure
Start at Cycle 60 Time = 2 ms

Result at Time = 6 ms

<table>
<thead>
<tr>
<th>CPUs</th>
<th>Initial Pressure</th>
<th>Result Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><img src="image1" alt="Initial 2 CPUs" /></td>
<td><img src="image2" alt="Result 2 CPUs" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image3" alt="Initial 4 CPUs" /></td>
<td><img src="image4" alt="Result 4 CPUs" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="image5" alt="Initial 8 CPUs" /></td>
<td><img src="image6" alt="Result 8 CPUs" /></td>
</tr>
</tbody>
</table>
Other Enhancements and Defect Corrections

Enhancements to FFCONTR to Initialize Air Pressure Inside Fluid-filled Bottles

The FFCONTR entry is extended to include \( P\text{INIT} \) variable to initialize the initial air pressure inside the bottles. In the previous releases, the air pressure inside the bottles was always assumed to be equal to the ambient air pressure given on the FFCONTR entry.

The example below demonstrates the pressure (MPa) of the air inside the bottle when the liquids cool. The red line uses \( P\text{INIT} = P\text{AMBIENT} \) and the blue line uses \( P\text{INIT} = 0.12 \).

![Graph showing pressure vs. time with red and blue lines demonstrating the effect of different initial pressures.]

Arbitrary Orientation of MESH, BOX

In previous releases, MESH, BOX could only create meshes that were aligned with the global coordinate axes. This imposed limitations on how the structure was positioned in the global coordinate system. To remove these constraints, a local coordinate system option has been added to MESH, BOX. As a result, the Euler mesh created by MESH, BOX can now have an arbitrary orientation. The following figure demonstrates the Euler mesh used in the slanted piston example in the Dytran Example Problem Manual. Here, the Euler mesh is defined by CHEXA's. The picture below shows the same Euler mesh created by MESH, BOX using the new local coordinate system option. This new option does not support PARAM, FASTCOUP.
Output Enhancements to Support Several “ALL” Specifiers for Material Time History Output

In previous releases, the time histories output for each material was supported through the MATS entry. However, it was not possible to output the total summaries for the material. In this release, the output requests are enhanced to generate material summaries for the following:

- ALLMAT for all material
- ALLEULMAT for all euler material
- ALLAGMAT for all lagrangian material

To check the total energy balance, the material variable $ETOT$ is also added and can be requested on the MATOUT entry. The following figure shows the total energy, total internal energy, and total kinetic energy for the slanted piston example. This example contains one Lagrangian and one Eulerian material.
Chapter 2: New Capabilities in HPC and Fluid-structure Interaction (FSI)  
Other Enhancements and Defect Corrections

The variable \( ETOT-ALLMAT \) shows the total energy conservation. The outflow of energy of Eulerian materials is not added to \( ETOT-ALLMAT \). So, if in a simulation, Eulerian mass flows out, then \( ETOT-ALLMAT \) will not remain constant.

**Marker Support for General Coupling**

Markers were only supported when using `FARAM, FASTCOUP`. They now support meshes using general coupling. In addition, at cycle 0, a summary of markers is given. This summary lists the marker coordinates, the element the marker is in, and the distance between marker and Euler element center.

**Faster Euler Archive Import**

An Euler archive of one run can be used for the initialization of a second run. The initialization of this remap run can take longer if the number of Euler elements is large. To address this, a sophisticated algorithm for computing the intersection between the imported Euler mesh and the new Euler mesh has been developed. This new algorithm can result in a substantial time saving of the initialization phase of the remap simulation.

**Easy Switching from EOSJWL to EOSGAM**

By using a negative time of the `DETSPH` entry, the detonation of EOSJWL material will be instantaneous and will become, in effect, an EOSGAM material. See the “JWL Material model vs. the Ideal Gas Approximation” example in Chapter 3: Fluid Dynamics of the *Dytran Example Problem Manual* for more details.

**Defect Corrections**

The following software defects are fixed in this release:

<table>
<thead>
<tr>
<th>Defect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYT-122</td>
<td>Issue a warning when the same PID appears on <code>CFACES</code> and <code>QUAD</code>: program exception - access violation</td>
</tr>
<tr>
<td>DYT-125</td>
<td>Solve memory leak that occurs in Roe solver when using multiple coupling surfaces</td>
</tr>
<tr>
<td>DYT-127</td>
<td>Element type 14 (real tet4) does not support <code>BODYFORCE</code></td>
</tr>
<tr>
<td>DYT-129</td>
<td>Submesh does not work when inserting fine mesh into example 4_6</td>
</tr>
<tr>
<td>DYT-131</td>
<td>Submesh does not work for multi-coupling surfaces</td>
</tr>
<tr>
<td>DYT-141</td>
<td>Very small uncovered fractions are eliminated causing deactivation of flow through porous boundaries</td>
</tr>
<tr>
<td>DYT-177</td>
<td>When using same elements for <code>COUPLE</code> and <code>MATINT</code> initialization of Euler is wrong</td>
</tr>
<tr>
<td>DYT-183</td>
<td><code>verif/press-extrapolation/cs-t9</code> shows diffs larger than 10% depending on running 1 or 2 CPUs</td>
</tr>
<tr>
<td>DYT-247</td>
<td>Time History still a problem in release</td>
</tr>
<tr>
<td>DYT-248</td>
<td>fsi model fails on 4 CPUs but is OK on 1 CPU</td>
</tr>
<tr>
<td>DYT-249</td>
<td>Viscosity and adaptive fluid mesh error</td>
</tr>
<tr>
<td>DYT-250</td>
<td>Trivial/t11/n11016porf on 8 CPUs terminates because of corrupt memory</td>
</tr>
<tr>
<td>DYT-251</td>
<td>$verif/graded-meshes/Blast-t11 coredumps on 8 CPUs</td>
</tr>
<tr>
<td>DYT-252</td>
<td>das error in verif/graded-meshes/nonuniform2 when running on 8 CPUs</td>
</tr>
<tr>
<td>DYT-253</td>
<td>Timestep too small for verif/gradedmeshes/nonuniform2 when running on 8 CPUs plus using second-order</td>
</tr>
<tr>
<td>DYT-254</td>
<td>$verif/multi-fvsf/EP4_7 hangs when running on 8 CPUs</td>
</tr>
<tr>
<td>DYT-255</td>
<td>$verif/roe/multi gives timestep too small on 8 CPUs</td>
</tr>
<tr>
<td>DYT-256</td>
<td>dt/trivial/fcoupling021 gives erroneous results on 2 CPUs</td>
</tr>
<tr>
<td>DYT-257</td>
<td>Trivial/fcoupling021 gives erroneous results on 2 CPUS</td>
</tr>
<tr>
<td>DYT-258</td>
<td>$verif/multi-fvsf/fvpor-ro2 on 16 CPUs tries to allocate memory of size 0</td>
</tr>
<tr>
<td>DYT-259</td>
<td>$verif/multi-fsv/fvpor7 gives das error on 16 CPUs</td>
</tr>
<tr>
<td>DYT-260</td>
<td>verif_multi-fsv_roe-fail1 on 16cpu shows write warnings</td>
</tr>
<tr>
<td>DYT-261</td>
<td>verif_graded-meshes_nonuniform gives das error on 16 cpus</td>
</tr>
<tr>
<td>DYT-262</td>
<td>$verif/multi-fvsf/fvpor2 cannot connect Euler meshes by porous holes on 32 CPUs</td>
</tr>
<tr>
<td>DYT-263</td>
<td>$verif/graded-meshes/nonuniform when run on 32 CPUs gives many irrelevant error messages</td>
</tr>
<tr>
<td>DYT-264</td>
<td>$verif/graded-meshes/Blast-3meshes on 32 CPUs gives das error</td>
</tr>
<tr>
<td>DYT-265</td>
<td>$verif/roe/multi gives timestep too small on 16 CPUs</td>
</tr>
<tr>
<td>DYT-266</td>
<td>$verif/coupling/n005 gives many messages after adding coupling surface output</td>
</tr>
<tr>
<td>DYT-267</td>
<td>VOID in single HYDRO definition with PORFLCPL</td>
</tr>
<tr>
<td>DYT-268</td>
<td>$verif/coupling/n005 gives timestep too small on 16 CPUs</td>
</tr>
<tr>
<td>DYT-269</td>
<td>Solve blocked flow for $verif/multi-fvsf/fvpor3 on 16 CPUs</td>
</tr>
<tr>
<td>DYT-270</td>
<td>$verif/coupling/n005 coredumps on 16 CPUs</td>
</tr>
<tr>
<td>DYT-271</td>
<td>$verif/graded-meshes/blast-roe gives das error on 16 CPUs</td>
</tr>
<tr>
<td>DYT-272</td>
<td>Manual edit results in Fortran crash</td>
</tr>
<tr>
<td>DYT-273</td>
<td>$verif/graded-meshes/nonuniform/gas7.dat on 32 CPUs terminates because there are too few elements</td>
</tr>
<tr>
<td>DYT-274</td>
<td>DMP does not work</td>
</tr>
<tr>
<td>DYT-275</td>
<td>$verif/example_manual/EP4_14 does not run on 2 CPUs</td>
</tr>
<tr>
<td>DYT-277</td>
<td>$verif/multi-fvsf/fvpor-ro3 uses coupling surface deactivation. This does not work well with DMP</td>
</tr>
<tr>
<td>DYT-278</td>
<td>THS file does not store Lagrangian results correctly when FSIDMP is activated</td>
</tr>
<tr>
<td>DYT-279</td>
<td>Problem with time history files created when DMP is employed</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>DYT-280</td>
<td>$verif/multi-fvsf/fvpor-roe gives faulty results on 4 CPUs</td>
</tr>
<tr>
<td>DYT-281</td>
<td>Discrepancies between 1 CPU and 4 CPUs run reported by baker oil tools</td>
</tr>
<tr>
<td>DYT-282</td>
<td>$verif/graded-meshes/strength coreumps for Windows 32 on two CPUs</td>
</tr>
<tr>
<td>DYT-283</td>
<td>Installation of DYTRAN cannot be “Read Only”</td>
</tr>
<tr>
<td>DYT-284</td>
<td>More than 1024 entities for output request SET causes an Error</td>
</tr>
<tr>
<td>DYT-316</td>
<td>Dytran DMP gives wrong euler initialisation with BIAS card</td>
</tr>
<tr>
<td>DYT-318</td>
<td>Dytran FSI-DMP is not working correctly</td>
</tr>
<tr>
<td>DYT-321</td>
<td>Input error for BODYFR1 using SURF and MATINI option</td>
</tr>
<tr>
<td>DYT-322</td>
<td>BODYFR1 doesn't work correctly</td>
</tr>
<tr>
<td>DYT-326</td>
<td>PORFLCPL error in DMP</td>
</tr>
<tr>
<td>DYT-330</td>
<td>verif/multi-fvsf/fvpor-roe2 fails HP DMP QA running 2 CPUs</td>
</tr>
<tr>
<td>DYT-331</td>
<td>ALLLAGSOLID not working for TET models</td>
</tr>
<tr>
<td>DYT-335</td>
<td>BOX1 and BOX with the same region do not give the same results</td>
</tr>
<tr>
<td>DYT-339</td>
<td>Euler archive contains multiple cycles (MESH with ADAPT) in Dytran2010 (wrong results)</td>
</tr>
<tr>
<td>DYT-340</td>
<td>dmp_verif_adap-output_fvpor fails on Windows32 DytranDMP with 2 CPUs</td>
</tr>
<tr>
<td>DYT-344</td>
<td>$verif/airbag/tvsurferT11 coredumps on 32 CPUs</td>
</tr>
<tr>
<td>DYT-345</td>
<td>$verif/example_manual/EP4_11 coredumps on 32 CPUs</td>
</tr>
<tr>
<td>DYT-347</td>
<td>verif/2nd_order/splash crashes on new machine with DAS corruption</td>
</tr>
<tr>
<td>DYT-348</td>
<td>verif/EulStrength/CouFric fails on linux32 with new compilers/OS for 2 CPUs</td>
</tr>
<tr>
<td>DYT-351</td>
<td>Dytran Example 6_5 coredumps on 4 CPUs</td>
</tr>
<tr>
<td>DYT-356</td>
<td>$verif/example_manual/EP4_12 gives &quot;grid point out of Euler&quot; error on 4 CPUs</td>
</tr>
<tr>
<td>DYT-357</td>
<td>DMP models are failing the regression tests on large cluster machine</td>
</tr>
<tr>
<td>DYT-358</td>
<td>Airbag models crashing on Solaris8664</td>
</tr>
<tr>
<td>DYT-359</td>
<td>verif_partitions_moving_tank crashing on Linus32</td>
</tr>
<tr>
<td>DYT-360</td>
<td>verif/example_manual/EP6_6 crashed in Dytran DMP on HPUX and AIX</td>
</tr>
<tr>
<td>DYT-363</td>
<td>PARAM, FASTCOUP, INPLANE does not work for porous segments that connect Euler domains</td>
</tr>
<tr>
<td>DYT-368</td>
<td>Using DMATEP for multi-material Euler gives strange message</td>
</tr>
<tr>
<td>DYT-369</td>
<td>Euler archive import takes too long for large Euler models because of quadratic loop</td>
</tr>
<tr>
<td>DYT-371</td>
<td>The explosive material model EOSJWL does not allow for instantaneous detonation</td>
</tr>
<tr>
<td>DYT-372</td>
<td>Markers do not work for non-orthogonal Euler meshes</td>
</tr>
<tr>
<td>DYT-373</td>
<td>Waste Basket does not work in Dytran Explorer</td>
</tr>
<tr>
<td>DYT-374</td>
<td>Intel compiler is the wrong version in Dytran Explorer</td>
</tr>
</tbody>
</table>
This PARAM with the YES option did not always work correctly for blended elements. This has been corrected in Dytran 2012. When using the PARAM with the YES option, there may be some small changes in results compared to results provided by Dytran 2010.
3 System Information

- Software Installation  24
- Licensing  24
- Release Platforms  25
- Memory Requirements  26
Software Installation

Dytran 2012 on Windows, Linux and Unix platforms can be downloaded from MSC.Software’s Solutions Download Center:

https://mscsoftware.subscribenet.com/

On the Windows platforms, Dytran 2012 can easily be installed as it uses the standard Windows Installation Wizard. On Unix and Linux platforms, the MSC.Software standard installation script can be used to install the software on your system. Dytran 2012 is the successor of Dytran 2010.

Licensing

Dytran uses the FLEXlm license manager as the licensing system for nodelock and network licensing.

To run Dytran, you need an authorization code from MSC.Software Corporation. If you already have a license for MSC.Dytran 2010, you will not need to obtain a new license for Dytran 2012. DMP capability is part of Dytran Standard and no additional licenses are needed to run DMP capability in Dytran 2012.

However, in all cases, you do need a new installation of the license server software. Specifically, the FlexLM license server needs to be at level 10 or higher. For this purpose, an installation of FlexLM v11.9 is part of this release on all supported platforms. It is noted that Dytran 2012 is not able to check out licenses when the FlexLM server is lower than version 10.

On Windows and Linux computers, Dytran requires an Ethernet card on your computer 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.
# Release Platforms

Dytran 2012 was built and tested on the following hardware with the listed software installed as given in Table 3-1.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Operating System</th>
<th>Compiler Version</th>
<th>OpenMP Support</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 32</td>
<td>Windows7</td>
<td>Intel Compiler Intel V 12.0*</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Windows x64</td>
<td>Windows7</td>
<td>Intel Compiler Intel V 12.0*</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>HP-UX PA-RISC 2.0</td>
<td>HPUX B.11.11</td>
<td>HP F90 V3.1</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>HP-UX Itanium2</td>
<td>HPUX B.11.23</td>
<td>HP F90 V2.8.7</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>Sun SPARC Solaris</td>
<td>Solaris 10</td>
<td>Sun Studio 12 (Sun Fortran 95 8.3 SunOS_sparc Patch 127000-01 2007/07/18)</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>Solaris x64</td>
<td>Solaris 10</td>
<td>Sun Studio 12 (Sun Fortran 95 8.3 SunOS_i386 Patch 127002-01 2007/07/18)</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>IBM RS/6000 (Power5)</td>
<td>AIX 5.3</td>
<td>XL Fortran 11.1</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>Linux Itanium2</td>
<td>RedHat 4 Update 5</td>
<td>Intel Compiler 10.1.012</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Linux Itanium2 SGI Altix</td>
<td>SGI ProPack5</td>
<td>Intel Compiler 10.1.011</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Linux X8664</td>
<td>RedHat 5 Update 4</td>
<td>Intel Compiler V 12.0</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
<tr>
<td>Linux 32</td>
<td>RedHat 5 Update 4</td>
<td>Intel Compiler V 12.0</td>
<td>Yes</td>
<td>Ethernet Card</td>
</tr>
</tbody>
</table>

*For correct operation of Intel Fortran Compiler, MS Visual Studio, NET 2010 must be installed prior to installing the Intel Compiler.
Memory Requirements

In general, the size of the memory required by Dytran depends on the size of the engineering problem you wish to solve. The default memory size is set to approximately 30MB. This default size is appropriate for smaller problems.

You can change the preset default in the 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.

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 Dytran Explorer, the settings are stored to be used the next time that you run the analysis. Under certain conditions, Dytran may stop and issue a message that it cannot allocate the required memory. Since the memory allocation in 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 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 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 to using Dytran on a computer that has at least 256 MB of RAM.
4 Using Dytran

- Running Dytran on Unix and Linux  28
- Running Dytran on Windows  28
- Postprocessing Dytran Results  29
- Postprocessing Dytran Results of Windows on UNIX  29
- Postprocessing Dytran Results with ParaView  29
- Time History Plots  45
Running Dytran on Unix and Linux

On Unix and Linux platform, you would use the command line interface like:

- `dytran jid=xxx` (to submit a regular Dytran job)
- `dytran jid=xxx bat=no` (to submit Dytran in interactive mode)
- `dytran jid=xxx exe=my_exe.exe` (to submit a Dytran job with a customized executable)

To submit DMP jobs, you must specify the number of processors as well:

- `dytran jid=xxx dmp=yes ncpus=2`
- `dytran jid=xxx dmp=yes ncpus=4 bat=no`

**Note:**
- `xxx` should be replaced by the name of your input deck without the `.dat` extension.
- Currently, it is not possible to create customized executables for DMP.

Running Dytran on Windows

On Windows, submit a Dytran analysis by double clicking the Dytran icon. The icon should be available on your desktop. Alternatively, you can use the Start Menu to locate Dytran under the Programs Folder. Once you picked either the icon or the menu entry, the Dytran user environment appears on your screen. The Dytran Explorer provides an on-line help system that contains information about the functionality of the Dytran Explorer. The Dytran Explorer provides some basic postprocessing and animation tools.

To submit DMP jobs, open a DOS command window and type:

```
{Full path to Dytran installation}\Dytran\2012\dytranexe\run_dytran.bat jid={job-name} dmp=dytran nproc={number of processors}
```

For instance:

1. Name input model is `bunker.dat`.
2. Dytran is installed at `C:\MSC.Software`.
3. Model needs to run on four CPUs in DMP mode.

The correct command would be:

```
C:\MSC.Software\Dytran\2012\dytranexe\run_dytran.bat jid=bunker dmp=dytran nproc=4
```

**Note:** Currently, it is not possible to create customized executables for DMP.
Postprocessing Dytran Results

Dytran results can be postprocessed with Patran. With Patran, the Direct Result Access (DRA) method is available for native Dytran output files (ARC,THS).

With Dytran 2012 there is also a translator available, called arc2vtk, that converts Dytran .ARC files into .vtu files that can be used as input for the open source data visualization program ParaView. ParaView can be downloaded from

http://www.paraview.org

In addition, on Windows, you can use the VisualVrml postprocessing, animation, and Visual Time History tool. The tool is built-in inside the Dytran Explorer and offers web-based postprocessing capabilities.

Postprocessing Dytran Results of Windows on UNIX

If you wish, you can postprocess the analysis results obtained from a Windows platform 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 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. On workstations with a little-endian architecture, 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 big-endian format (“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 big-endian UNIX-format files by the ux prefix.

For Patran and the Arc2Vtk translator the big-endian (“Unix”) to little-endian (“Windows”) conversions are not required, since they can handle both formats and do the conversion while reading the .ARC files.

Postprocessing Dytran Results with ParaView

How to obtain ParaView

ParaView is a free open source data visualization program developed by Kitware Inc. It is based on the Visualization Toolkit (VTK), also developed and maintained by Kitware Inc. ParaView can be downloaded from http://www.paraview.org. The installation is straightforward.
**Arc2Vtk Translator**

There are a number of readers available in ParaView that can handle a various data formats. MSC has developed the Arc2Vtk translator to convert the Dytran archive files into “VTK” files that can be read by the VTK-toolkit, which is part of ParaView.

In the case of Dytran ARC files, the “unstructured grid” variant of VTK with file extension `.vtu` has to be used, either in ASCII or base64 encoded binary format. The `.vtu` file format itself does not support cycle or problem time information, but this can be provided to ParaView by a meta-file with file extension. PVD that describes sets of `.vtu` file. The Arc2Vtk translator creates them automatically. For this purpose, time and cycle numbers are written in the VTU files as XML-comment.

ParaView can also create time history plots from a.o. “comma separated values” files. Arc2Vtk has an option to create these `.CSV` files from Dytran Time History files (`.THS`).

There is no graphical user interface available yet for Arc2Vtk, so the program has to be run from the command line. On Windows, you need to open a cmd window (`Start/All Programs/Accessories/Command prompt`); on Linux/Unix, you need to open a “terminal”. The easiest way to translate a set of Dytran archive files and prepare an appropriate PVD file is:

```
arc2vtk.exe my_file_name_XXX.ARC
```

with `my_file_name_XXX.ARC` the name of an archive file as created by Dytran. Arc2Vtk searches the current working directory for all `.ARC` files with the same prefix, but different cycle numbers ‘XXX’, translate all of them to VTU files and create a PVD file called “my_file_name_XXX.pvd”.

The PVD file can be opened in ParaView.

To translate a single Dytran/SOL700 archive file use:

```
arc2vtk.exe [-binary|-ascii] -nonauto my_file_name.ARC
```

Arc2Vtk creates one or more `.VTU` files, depending on the number of time steps found on the archive file. The same naming convention is used as in Dytran, which means that when cycle `XXX` is found on archive file `my_file_name_YYY.ARC`, the corresponding output file will be named `my_file_name_XXX.VTU`.

It is strongly recommended to use the default settings: `-xml` and `-binary`.

To create a PVD-file from a set of VTU files already created by Arc2Vtk use:

```
arc2vtk.exe my_file_name_XXX.vtu
```

This will create a PVD-file for the collection of all VTU-files matching the name `my_file_name_XXX.vtu`, where `XXX` are cycle numbers. Note that the argument should be the name of an existing VTU file.

The resulting output file is named `my_file_name_XXX.PVD`, so for example `MY_FILE_0.VTU` results in the output file `MY_FILE_0.PVD`.

To create a set of CSV-files (comma separated values) from a Dytran time history file (`THS`) use:

```
arc2vtk.exe my_file_name.THS
```
This creates a CSV-file for each of the “items” (gridpoint, element, face or material, depending on the type of THS-file) on the time history file, named my_file_name_X_NNN.csv. With “X” replaced by the type of item (N for node, E for element, F for face and G for global data) and NNN its identifier number. CSV files can be processed by ParaView, but also be used as input for spreadsheet programs like Excel.

Running ParaView

The user interface of ParaView is rather intuitive and creation of a simple animation is not too difficult. The program has a lot of capabilities and describing all of them here is not possible. Furthermore, there is also a Python interface.

To startup ParaView go to Start/Programs as usual.

In the following example we will demonstrate:

• How to open files in ParaView
• How to select results variables to display
• How to apply a clipping plane
• How to open a series of VTU (translated .ARC) files
• How to apply a filter to convert cell data to point data
• How to create an isosurface
• How to make an object transparent
• How to start an animation
• How to save an animation
1. Open a file

One can open either single .vtu files (click on the “+” sign to expand the list when there are more .vtu files with the same base name), a whole series of .vtu files by selecting the first of the list without expanding it, or the preferred way, by selecting the .pvd meta-file. When the results of more than one series of archive files have to combined, for example when there are separate Euler and Lagrange data, using the .pvd file is the only way to get the results synchronized.

- Click on file in the menu bar at the top of the ParaView window.
- Click Open.

![ParaView 3.10.1 32-bit](image-url)
• Use the Explorer window to browse to the file(s) you want to open and click OK. The name of the file will appear in the Pipeline Browser.

The file name will appear in the Pipeline Browser on the left side of the ParaView window. You need to click Apply on the Properties tab of the Object Inspector pane that is positioned below the Pipeline Browser.

Reading and processing of the data may take some time.
• Click on Apply on the Properties tab of the Object Inspector.
2. Positioning of the object

You can use the buttons with the axis icons in the toolbar to position the object in the display pane. Furthermore, clicking one of the mouse buttons and moving the mouse when the cursor is in the display window will enable you to:

- left mouse button: rotate the object
- middle mouse button: move the object in horizontal and vertical directions
- right mouse button: zoom in and out
- scroll wheel: zoom in and out
3. Displaying results

To create fringe plots of results:

- Make sure the object on which you want to make the plot is selected in the Pipeline Browser.
- Open the Display tab of the Object Inspector and select the variable you want to use for the fringe plot.
- Select the variable you want to display.
- Open the Properties tab of the object inspector and click the (green) Apply button.
To change the color map:

- Open the Display tab of the object browser.
- Click Edit Color Map.
- Click either Rescale to Range Data if you want to apply the color map to the current time step only, or click Rescale to Temporal Range if you want a color map that will cover the range in all time steps. The latter may take some time.
- Click Close when finished.
4. Applying a clipping plane.
   To look inside objects, you can apply one or more cutting planes.
   • Make sure the object on which you want to apply the cutting plane is the active one in the Pipeline Browser.
   • Click the Cutting Plane icon on the toolbar.
   • Position the cutting plane, either with the buttons and data fields on the Properties tab of the Object Browser or by manipulating the origin of the cutting plane, its normal or the borders with the mouse cursor in the display window.
Click Apply when finished.
5. Opening more than one series of results files.

Opening a second or higher series of results files, is similar to the way the first series was opened, just use File/Open on the menu bar and browse to the .pvd file you want to add to your visualization.

The file appears as another object in the pipeline browser.

6. Creation of contour plots

Most Dytran results are zone variables, that are translated to VTK “Cell data” and will look like a pile of blocks in ParaView. They can be examined with cutting planes, but most of the time you will also want to make contour plots on isosurfaces, vector plots or generate streamlines. Unfortunately, this can only be done with “Point data” in ParaView. However, there is a filter that can transfer “Cell data” into “Point data”.

- Make sure the object on which you want to apply the filter is selected in the Pipeline Browser.
- Click on Filters in the menu bar.
- Click Alphabetical.
- Click Cell to Point Data.

The translation of the “Cell data” and the creation of the “CellDatatoPointData” object may take some time.
Now we can create contour plots:

- Make sure the newly created CellDatatoPointData object is selected.
- Click on the Contour icon on the toolbar.
- Select the variable to be used for the creation of the isosurface in the Object Selector (here FMATPLT), use New Value to create the value for which you want to create the isosurface (here 0.5) and use Delete to remove the value Paraview did set as default (here 0).
- Click Apply to create the isosurface
7. Transparency

The combination of mesh plot of the structure with a cutting plane and the iso-surface of the Euler mesh does not give information that is very clear in this situation.

You can apply cutting planes on the iso-surfaces as well and making them transparent.

- Select the object of the structure.
- Lower the value for Opacity on the Display tab of the Object Inspector.
8. Animation

Use the “animation” buttons on the menu bar to play/stop animations of series of timesteps.

The functions of the buttons are from left to right:

- Jump to first frame
- Go one frame backward
- Start/Stop the animation
- Go to the next frame
- Jump to the last frame
- Loop
It is also possible to save animations as AVI files:

Click File on the menu bar and select Save Animation.

This will bring up the Animation Settings Dialog where you can set things like Frame Rate (frames per second), No. of Frames / timestep, and resolution (pixels). For smaller numbers of time steps (10-25), the defaults look appropriate. If you have larger numbers of timesteps (say 100-500), you could consider to increase the number of frames per second. For professional videos, 12.5 (PAL) or 15 (NTSC) are appropriate values. Also, be aware that larger is not always better since many media players cannot handle large or/and non-standard sized frames without hickups. Standard PAL (720 x 576) or NTSC (640 x 480) are safe choices if you want to be sure that the result will be playing smoothly on any ones computer.
Animations can be saved as AVI (unfortunately this AVI variant is not the most optimal with respect to file size), or separate frames in JPEG, PNG or TIFF format, that can be used by third party products like ImageMagick to create MPEGs.

ParaView cannot read Dytran Time history files (THS) directly. They need to be translated to CSV files (comma separated value) first by arc2vtk.

Use:

```
arc2vtk.exe my_file_name.THS
```

To create one or more CSV-files (comma separated values) from a Dytran time history file. For each of the “items” (gridpoint, element, face or material, depending on the type of THS-file), a CSV-file will be created named `my_file_name_X_NNN.csv`, with `X` replaced by the type of item (N for node, E for element, F for face, and G for global data) and `NNN` its identifier number. CSV files can be processed by ParaView, but also be used as input for spreadsheet programs like Excel.

### A. Opening CSV-files in ParaView

To open a CSV-file in ParaView use “File / Open” in the menu bar
Opening CSV-files in ParaView

To open a CSV-file in ParaView use File/Open in the menu bar:
This will pop-up an explorer window where you can select the file to be opened as usual. When there are multiple files with similar file names, only differing in a number, they will be displayed as a series. To select a single file, you need to expand the list first by clicking on the “+”-sign.
For some reason, ParaView is aware of `.csv` being the extension of a supported file type, but it apparently doesn't know, what reader has to be used. Select the Comma-separated-values reader in the window that pops-up.
The name of your CSV file will be listed in the Pipeline Browser, but you still need to click the Apply button, to start the actual read.
The contents of the CSV file are being displayed as a spreadsheet:

<table>
<thead>
<tr>
<th>Time</th>
<th>Pressure</th>
<th>XEL</th>
<th>XYL</th>
<th>ZYL</th>
<th>AREA</th>
<th>Row ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.000109248</td>
<td>1.2731e-06</td>
<td>0.856081</td>
<td>-4.9849e-06</td>
<td>86</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.00003941</td>
<td>1.01261</td>
<td>2.35628e-06</td>
<td>0.648881</td>
<td>-4.7294e-06</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>0.00031229</td>
<td>1.01165</td>
<td>1.04e-06</td>
<td>1.33052</td>
<td>-1.3953e-05</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>0.00001002</td>
<td>1.01041</td>
<td>3.5196e-06</td>
<td>1.61112</td>
<td>-1.6957e-05</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>0.00051597</td>
<td>1.00665</td>
<td>3.7659e-06</td>
<td>1.92661</td>
<td>-5.666e-05</td>
<td>86</td>
</tr>
<tr>
<td>6</td>
<td>0.0006039</td>
<td>1.0037</td>
<td>3.6744e-06</td>
<td>2.12571</td>
<td>-7.2600e-05</td>
<td>86</td>
</tr>
<tr>
<td>7</td>
<td>0.00019632</td>
<td>1.0025</td>
<td>3.2130e-06</td>
<td>2.2549</td>
<td>-4.4492e-05</td>
<td>86</td>
</tr>
<tr>
<td>8</td>
<td>0.000019632</td>
<td>1.00293</td>
<td>2.6018e-06</td>
<td>2.2541</td>
<td>-5.7021e-05</td>
<td>86</td>
</tr>
<tr>
<td>9</td>
<td>0.009001</td>
<td>1.0012</td>
<td>1.9132e-06</td>
<td>2.49721</td>
<td>-3.9028e-05</td>
<td>86</td>
</tr>
<tr>
<td>10</td>
<td>0.0101456</td>
<td>1.00023</td>
<td>6.8437e-07</td>
<td>2.3867</td>
<td>-5.8868e-05</td>
<td>86</td>
</tr>
<tr>
<td>11</td>
<td>0.0110511</td>
<td>9.99989</td>
<td>-2.72076e-07</td>
<td>2.33049</td>
<td>-6.0021e-03</td>
<td>86</td>
</tr>
<tr>
<td>12</td>
<td>0.0112102</td>
<td>9.97695</td>
<td>1.42234e-06</td>
<td>2.22226</td>
<td>-4.0001e-03</td>
<td>86</td>
</tr>
<tr>
<td>13</td>
<td>0.0130062</td>
<td>9.9745</td>
<td>2.22267e-06</td>
<td>2.1151</td>
<td>-6.0002e-03</td>
<td>86</td>
</tr>
<tr>
<td>14</td>
<td>0.0142859</td>
<td>9.96767</td>
<td>2.95184e-06</td>
<td>1.9652</td>
<td>-6.0002e-03</td>
<td>86</td>
</tr>
</tbody>
</table>

To create XY-graphs, you need to apply a “Plotdata” filter:

- Open the filter menu on the menu bar on top of the ParaView window
- Select the Data Analysis submenu and select Plot Data
In ParaView, the Apply button has to be clicked to start the actual work.
Now the screen looks like this:

The usual Pipeline Browser and Object Inspector on the left, a spreadsheet in the middle, and a narrow XY-graph at the right.
The graph contains curves of all variables found on the CSV file, plotted against the same scale. This is most likely not what you want. First deselect the variables you do not want in the current graph:

- Make sure the window with the graph is selected (border colored blue).
- Open the Display tab on the Object Inspector.
- Deselect the variables you don't want.
- Probably you also want to change the X-Axis to time:
  Select Use Data Array
  Sometimes it's necessary to toggle variable time on and off to force a redraw of the graph.
The ParaView window will look like this:
Use the Resize Window buttons at the right top of the graph window to expand it to full size.

Giving the final result below.

You can play with the size and aspect ratio of the Paraview window to get the format you want.

Use the Resize Window button on the top right to go back to the initial situation.
Adding text to XY-plots

Click the View Settings button on the left side on top of the graph window to add headers or/and axis labels to a plot or to change the appearance of the axes.

This brings up the View Settings menu where you can add a title or add labels to the axes.
With Layout, it is also possible to define the axis spacing manually or switch to logarithmic scale.
The final result may look like this:

![Graph showing multiple CSV files combined]

**Processing Data from Multiple CSV Files**

It is possible to combine the results from multiple files in one graph.

- First create a graph with the results of one CSV file as described above.
- Also open the second file and apply a PlotData filter to it.
- Make sure the window with the graph of the first file is active (blue border).
- Select the second PlotData filter in the Pipeline Browser and change the settings on the Display tab of the Object Inspector.
- The curve also appears in the active window.
- Use Choose Color.. to open a menu where you can change the color of the curve of the selected (blue background) variable.
Chapter 4: Using Dytran

Time History Plots

Pipeline Browser

Object Inspector

Line Series

Choose Color...
The final result may look like this: