HypDev and HypVisual
Reference manual

Hyperion Project "Physical simulation in real-time"	
Version	0.7
Reference	DR 11-05-01
Creation	Pierre Rebours – 11/05/01
Updating	Pierre Rebours – 15/06/01
Updating	Pierre Rebours – 19/09/01 – Remove resources no related to the Client Applications.
Context	Basic How To
Objectives	Explains how to use the Client Applications of the project. Advice about some FEM techniques.
Classed	Public

Hyperion Project

Summary

Preamble

The most recent updates concerning documentation, binaries and source code can be found on the web site of the project at the following address:
http://lagrandeporte.multimania.com/hyperion

1 HypDev

Introduction

HypDev is a console application used to calculate the linear relations between the displacements of rigid bodies and applied forces. To achieve its aim, the application uses an adaptation of the Finite Element Method (FEM) christened “Hyperion Method”. For more information, refer to the technical articles related to the project.

The next step, which is in real time, is handled by the *HypVisual* application. See the next chapter for further information.

The goal of *HypDev* is to find out the following linear relation:

$$U=K^{-1}F$$

with U , the displacement vector and F , the force vector.

Let us exanimate the different steps to find this relation. The standard output notifies the user when a step is carried out.

Step 1, Meshing of the body

A division of the body is done into finite elements; this step is called *meshing*. At the actual state of the implementation, *HypDev* could only achieve the divisions into cubic elements. You must choose a trade-off between the density of elements and the computation time to obtain results more or less accurate.

During the construction of the linear relations between the forces and the displacements, the computation time is $O(n^3)$ with n the number of free nodes.

During the execution of the level, the time computation is maximum O(n) because we only carry out a multiplication between a matrix and a vector.

Parameters

These parameters are regrouped under the label `-g` (g like Geometry)

Numbering	*Type*	*Description*
`1`	`double`	Dimension of the cube along the X axe
`2`	`double`	Dimension of the cube along the Y axe
`3`	`double`	Dimension of the cube along the Z axe
`4`	`int`	Number of elements along the X axe
`5`	`int`	Number of elements along the Y axe

6	int	Number of elements along the Z axe

Example

```
-g 5 10 20 2 4 8
```

A cube is created with a width of 5, a height of 20 and a depth of 20. The number of elements is 64 (2x4x8). The number of vertices (which rely the elements together) is 135 ([2+1] x [4+1] x [8+1]).

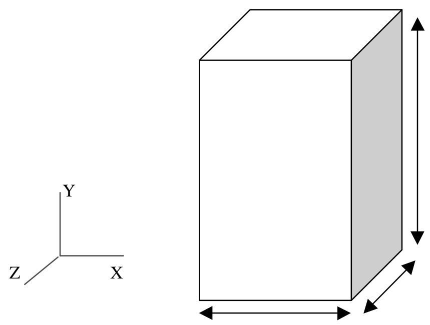


Fig. 1 Body generated in the example

Step 2, Construction of the global matrix

When the cube is divided into small cubic elements, the construction of the global matrix can begin. At this time, all the elements are associated with a same material, called default material. Two numbers identify the property of this default material.

	E (Elastic modulus in Pa)	*Sigma (Poisson modulus)*
Default material	100 000	0.2
Steel	240e6	0.2

The following relation is found:

$F = KU$, K is called the global matrix.

Since no extra-parameter is necessary, the construction of the global matrix follows the meshing step.

Example

```
-g 5 10 20 2 4 8
```

The dimension of the global matrix is equal to the number of nodes multiply by the numbers of DOF (i.e. 3). If the “g” parameters are 5 10 20 2 4 8, the dimension is 192 (i.e. 2*4*8*3) by 192.

Step 3, Construction of the boundary matrix

Each physical problem is associated with boundary conditions. By the way, a finite element problem has boundary conditions as well. In this project, we only focus on displacement conditions.

The size of the global matrix is reduced after applying the displacement conditions. The new matrix is called *Boundary Matrix* and noted K'.

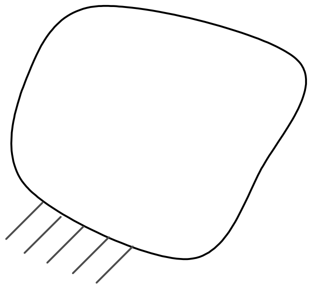


Fig. 2 This body is fixed on its base

Of course, the final solution will change if the base of cube is fixed or if the cube simply lies on a ground.

HypDev handles the boundary conditions with a set of parameters following the `–dof` label. Several `-dof` labels could be used.

Parameters

These parameters are regrouped under the label `-dof` (dof like Degree Of Freedom). The six first parameters allow selecting nodes, which are inside a specific region.

Numbering	*Type*	*Description*
1	`double`	If `P[4]` is true, DOF are applied on nodes in the plan which has `x=P[1]` for equation
2	`double`	If `P[5]` is true, DOF are applied on nodes in the plan which has `y=P[2]` for equation
3	`double`	If `P[6]` is true, DOF are applied on nodes in the plan which has `z=P[3]` for equation
4	`bool`	Select the x plan
5	`bool`	Select the y plan
6	`bool`	Select the z plan
7	`bool`	Fix the DOF of the selected nodes along the x axe
8	`bool`	Fix the DOF of the selected nodes along the y axe
9	`bool`	Fix the DOF of the selected nodes along the z axe

Examples

DOF are applied on nodes that satisfied all geometrical conditions. For example,

```
-dof 0 0 10 0 0 1 1 1 1
```

HypDev fixes the degree of freedom in the three direction of all the nodes which are in the plan z=10.

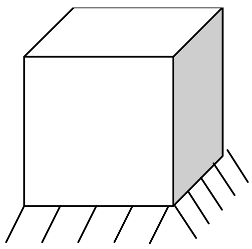


Fig. 3 Body fixed at it base

The two notations are equivalent:

```
-dof 0 0 10 0 0 1 1 1 1
```

```
-dof 50 20 10 0 0 1 1 1 1
```

because the first and the third parameters are ignored : the 4th and the 6th parameters are false.

```
-g 10 10 10 2 2 2 -dof 0 0 20 0 1 1 1 1 1
```

This notation fixes no degree of freedom of the body because nodes, whose the coordinates are Y=10 and Z=0, don't exist.

```
-dof 0 10 10 0 1 1 0 1 1
```

HypDev fixes the degree of freedom along the y and the z-axes of all the nodes which have Y=10 and Z=10 for coordinates.

With the finite element method, the application of the boundary conditions are a very important and delicate stage. One of the most important rules is to fix enough the structure to prevent its movement. For example, if the parameters are

```
-dof 0 0 0 1 1 1 1 1 1
```

the result could be unexpected and surprising, because the structure is only fix with one node. So it could turn along the three axes.

Step 4, Inversion of the boundary matrix

Enough degrees of freedom have been applied on the body, so the rank of the matrix has decreased. With numerical methods, we always find the inverse of a no reversal matrix. The solution to known if a matrix is reversible is not to calculate it determinant but to calculate its conditionment. More the conditionment is important (the ideal case is around one), more the computation of the inverse of a matrix is sharp.

The matrix obtained is K'^{-1} . This is the longest step.

Step 5, Render simplification

Some nodes are invisible but necessary to construct all the finite elements. Now, theses nodes are useless. The matrix is again simplified to obtain a new matrix called Render Matrix.

Step 6, file generation

It is time to save the linear relation between forces and displacements. The result is save in a Hyperion file. This file is an extension of the DirectX file format because new templates are defined.

```
template HypMatrix {
 <af8b31e1-36a0-11d5-a099-0080ad97951b>
 DWORD nRows;
 DWORD nColumns;
 array FLOAT Matrix[nRows][nColumns];
}

template HypNode {
 <5151b1c2-3f11-11d5-a099-0080ad97951b>
```

```
 DWORD Label;
 FLOAT X;
 FLOAT Y;
 FLOAT Z;
 DWORD Properties;
 DWORD DOF;
}

template HypBase {
 <5151b1c5-3f11-11d5-a099-0080ad97951b>
 DWORD Label;
 DWORD nRefNodes;
 array DWORD RefGlobalNodes[nRefNodes];
 array DWORD RefLocalNodes[nRefNodes];
}

template HypElement {
 <5151b1c6-3f11-11d5-a099-0080ad97951b>
 HypBase Base;
}

template HypSide {
 <5151b1c7-3f11-11d5-a099-0080ad97951b>
 HypBase Base;
}

template HypMesh {
 <5151b1c1-3f11-11d5-a099-0080ad97951b>
 DWORD nNodes;
 array HypNode nodes[nNodes];
 DWORD nSides;
 array HypSide sides[nSides];
 DWORD nElements;
 array HypElement elements[nElements];
}
```

Parameters

These parameters are regrouped under different labels.

Label	*Parameter*	*Description*
`-hypfile`	File path	Where is save the linear relation. Native format of the Hyperion module.
`-xfile`	File path	Where is save the geometry of the file under the DirectX format.
`-txtfile`	File path	Where is save textual information about the geometry, the degrees of freedom of the rigid body.
`-logfile`	File path	Where is save log information generated during the execution of the application.

Example

```
-xfile c:\test.x -hypfile c:\test2.hyp -txtfile c:\test3.txt
```

Save the rigid body in the three possible formats.

Further information

Note that only the Hyperion Format is useful for the next application *HypVisual*.

With *HypDev*, there is an easy way to check the validity of the founded linear relation:

- Apply forces;
- See the deformation with the DirectX file (`-xfile` label) associated with a external DirectX file viewer
- Check out the numerical value with the text file (`-txtfile` label).

To apply this force, we use the label `-f` that works like the `-dof` label.

Numbering	*Type*	*Description*
1	double	If `P[4]` is true, forces are applied on node in the plan which has `x=P[1]` for equation
2	double	If `P[5]` is true, forces are applied on node in the plan which has `y=P[2]` for equation
3	double	If `P[6]` is true, forces are applied on node in the plan which has `z=P[3]` for equation
4	bool	Select the x plan
5	bool	Select the y plan
6	bool	Select the z plan
7	double	Force of the selected node along the x axe
8	double	Force of the selected node along the y axe
9	double	Force of the selected node along the z axe

Common cases

HypVisual 2

File format

The Hyperion Format is the only one accepted by this application. For further information about this format, see the section about the *HypDev* application.

Several files have been generated with the *Hypdev* application. We could find these files in the path called `doc/ex_files`. By default, *HypVisual* loads the file called `default_file.hyp, which` must be found in the same path as the application.

Let us see the functionality of the application through this default file.

Keyboard command

To see all the details of the loaded body, use the keyboard keys:

- To move sideways along the three axes X,Y,Z use the keyboard arrows and the +/- keys ;
- To rotate along the three axes, use the same keys but in addition press on the "Space" key.

A system of three axes is display on the screen.

Color of the axe	*Axe*
Green	X
Blue	Y
Red	Z

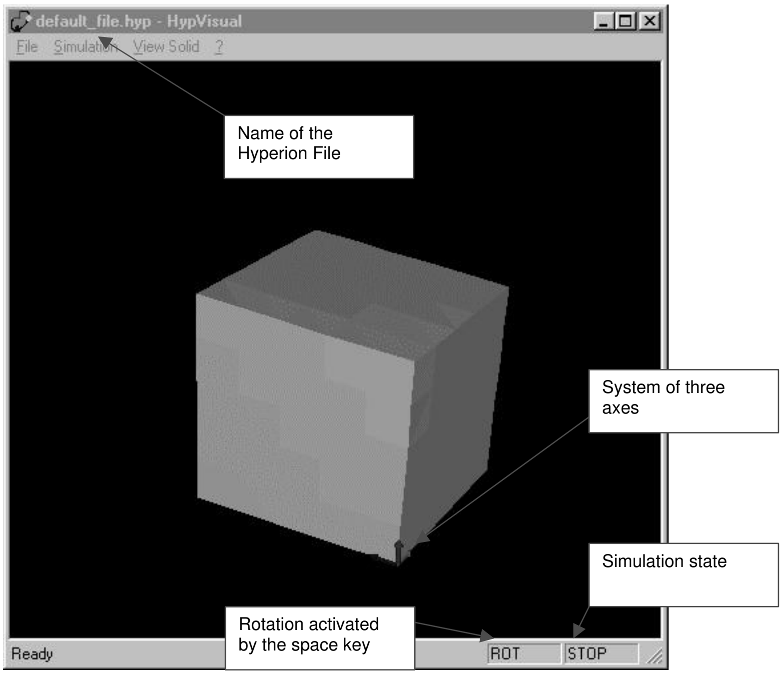
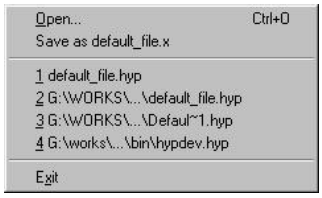


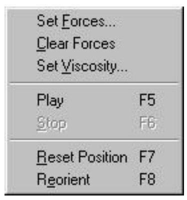
Fig. 4 Solid in its neutral position

File Menu



Open	Open a Hyperion File
Save as	Save the shape of the body under the DirectX format. This generated file could be viewed with *Direct3D RM Viewer* that is an application from the DirectX SDK.
Previously opened files	The list of the previously loaded files is inserted in the File Menu
Exit	Exit of the application

Simulation Menu



This is the most important menu of the application. It permits to apply forces on the solid and defines the parameters for the temporal interpolation.

Set Forces

Applies forces on the body. Two types of forces can be applied:

- Punctual forces;

- Pressure.

The principle is the same as the HypDev application:

- Definition of the force vector: *Force components;*
- Definition of the region where the force vector will be applied: *Region selection*.

Moreover, it is always important to check if the forces are really been applied. To do that, look at `Show Global Forces` function (see `Show Solid Menu`). Remember that force can be applied only on the nodes.

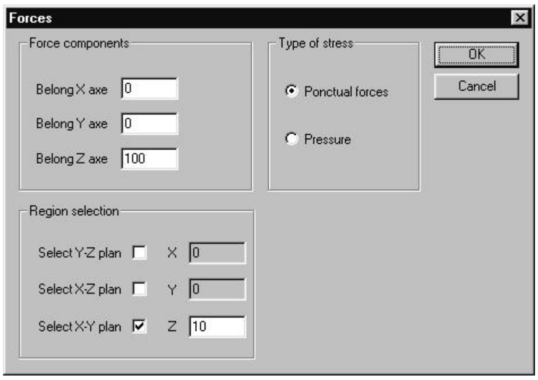


Fig. 5

According to the last figure, we apply a force F(0,0,100) on all the nodes that have z coordinate equal to 10.

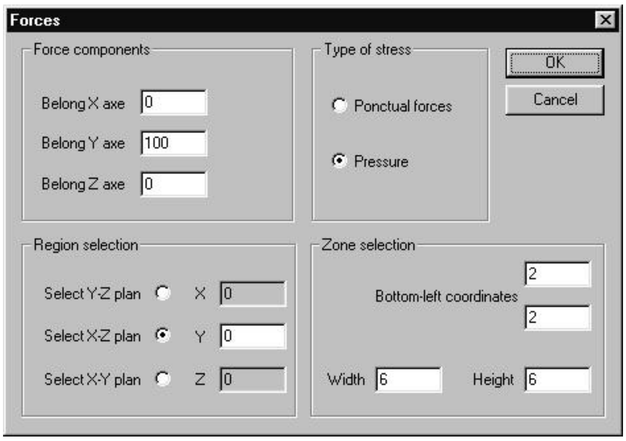


Fig. 6

Here is applied a pressure. The localization is a rectangle described in the *Zone Selection*. The result is showed below.

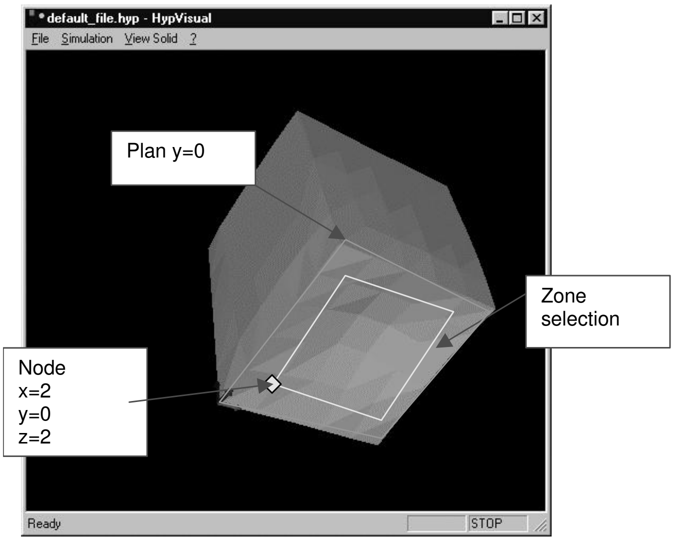
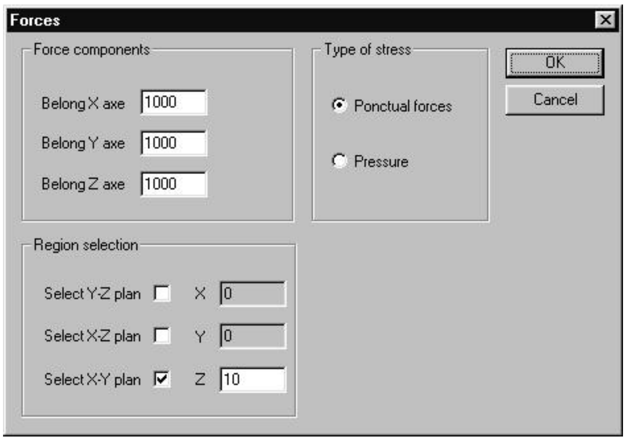


Fig. 7 Result of the simulation after an applied pressure

Notice that you could add the forces if you use several times the menu "Forces".

Warning

If you apply some excessive forces, like in the following example :



you will have some aberration. It is not a divergence of the method : the application only does a multiplication of a matrix by a vector. Remember that we work in the elastic domain. In the reality, a plastic domain replaces this elastic domain, so the mechanical comportment of the rigid body is different.

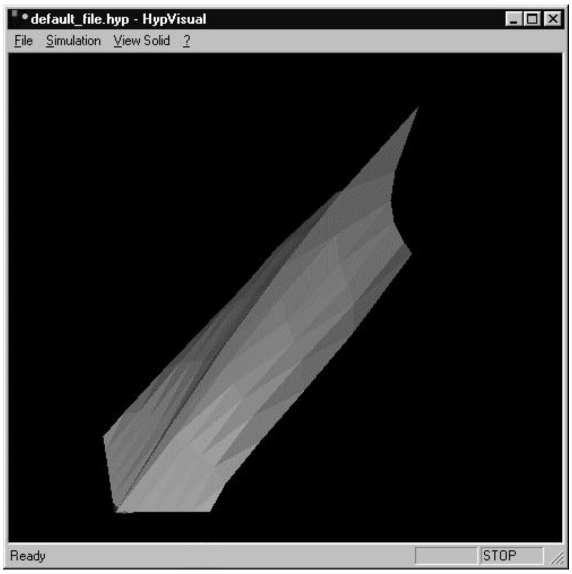


Fig. 8 A too important force has been applied

Set viscosity

Defines all the temporal interpolation parameters.

Clear forces	Suppress all the forces applied on the body
Play, Stop, Pause	Common controls designed for the temporal animation
Reset position	After a reset, the body returns to it previous form
Reorient	Reorient the solid to its initial state

Show Solid Menu

It is reserved for advanced users. With this menu, some lists are generated.

Nodes
Lists the nodes of the solid.

Sides
Lists the visible sides of the solid.

Elements
Lists the elements of the solid.

Matrix
Shows the dimension of the matrix located in the Hyperion file.

Global Displacements
Shows the displacements of each node.

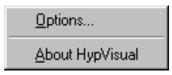
Local Displacements
Shows the displacements of each node - No documented.

Render Mask
No documented.

Global Forces
Shows the forces applied on each node.

Local Forces
Shows the forces applied on each node - No documented.

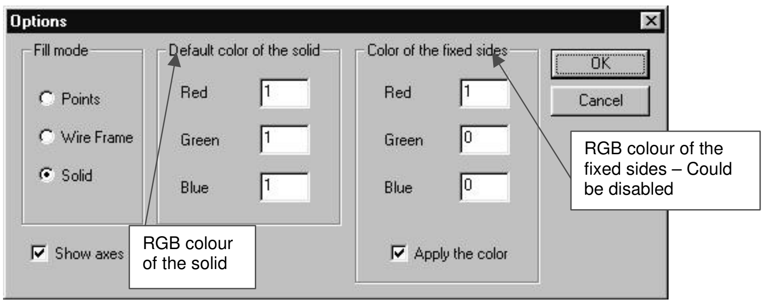
Force Mask
No documented.



? Menu

Options

Modifies graphical options.



About HypVisual