LS-INGRID 3.5

Graphical User Interface Manual

August 1998

<u>Mailing Address:</u>

Livermore Software Technology Corporation
2876 Waverley Way
Livermore, California 94550-1740

<u>Support Address:</u>

Livermore Software Technology Corporation
97 Rickenbacker Circle
Livermore, California 94550-7612

FAX: 925-449-2507
TEL: 925-449-2500
EMAIL: sales@lstc.com
WEBSITE: www.lstc.com

TABLE OF CONTENTS

TABLE OF CONTENTS

TEXT CONVENTIONS USED IN THIS MANUAL

TYPEFACES: - Three different typefaces are used in this manual:

Manual text This typeface is used for text in this manual.

Computer type

```
This one is used to show what the computer types.
It is also used for equations etc.
```

Operator type ***`This one is used to show what you must type.`***

NOTATION: - Triangular, round and square brackets have been used as follows:

Triangular To show generic items, for example:

`<component name>    <filename>    <angle>`

And also to show special keys, for example:

`<return>  <control Z>   <escape>`

Round To show optional items during input, for example:

`<command>  (<optional command>)  (<optional number>)`

And also to show defaults when the computer prompts you, e.g.:

```
Give new value (10) :
Give part name (UPPER TORSO) :
```

Square To show advisory information at computer prompts, e.g.

```
Give terminal type [M for list] :
```

Also to show implicit commands, e.g.

`[BLANK]  MAT  <list>`

1.0 INTRODUCTION

LS-INGRID Version 3.5A (Feb.1998) is based on the original INGRID code by Stillman, Hallquist, et al.

It contains all the "index space" mesh generation capabilities that traditional INGRID users employ, and has had added many new interactive features. The main ones being:

- A screen based, menu-driven interface for the interactive parts of the program;
- Improved graphics: both 2D (X-Windows) and 3D (SGi GL and HP Starbase);
- Interactive model merging and manipulation capabilities;
- Interactive occupant (dummy) positioning features;
- Interactive airbag folding;
- Interactive definition and positioning of seat-belts and related entities;
- Interactive definition and display of many model boundary conditions.

In addition the capabilities of the code have been extended so that it can act as a model editor and pre-processor to LS-DYNA3D. In particular it can:

- Read all LS-DYNA3D input file formats, including LS930 Keyword;
- Read some proprietary pre-processor formats: SDRC I_DEAS universal files, and PDA PATRAN neutral files;
- Read NASTRAN bulk data format;
- Multiple input decks from any or all of these sources may be merged into a single model, the parts of which may be rotated, scaled and translated interactively. The resulting composite file may be written out in a range of LS-DYNA3D formats.

This document summarizes how to use the capabilities described above, and it gives details of the special occupant features. However users wishing to use the "index space" mesh generation capabilities of INGRID are referred to the document:
"LS-INGRID, Commands and Reference Manual A Pre-Processor and Mesh Generator for LS-DYNA3D, LS-NIKE3D and TOPAZ3D", Version 3.5, February 1998.

2.0 RUNNING LS-INGRID

On most systems LS-INGRID will be run by the command:

```
INGRID
```

Although your system may have been customized in some other way.

2.1 Command line options

LS-INGRID may have any combination of the following command line options added to it:

`i=<input filename>` Defines the "ingrid" format filename to be used in a batch mode run. If this filename is defined then batch mode, i.e. traditional non-interactive usage, is assumed. The "ingrid" file will be executed and, depending on the commands in it, graphics may or may not be used.

`c=<input cfilename>` The Command Session File is saved after every termination of LS-INGRID in “isave”. This file contains all the commands used in the previous session. This file must be renamed to use as a script to recreate your last session, which can be invoked when executing LS-INGRID, or during an LS-INGRID session.

`o=<output filename>` Defines the name of the output filename, which defaults to **`ingrido`** if it is not given. Defining this parameter does not imply "batch" mode usage.

`<graphics device>` Defines the graphics device to be used for interactive graphics. If this is omitted it will be prompted for. (See 2.2 below.)

Therefore a typical LS-INGRID session might be invoked with the command:

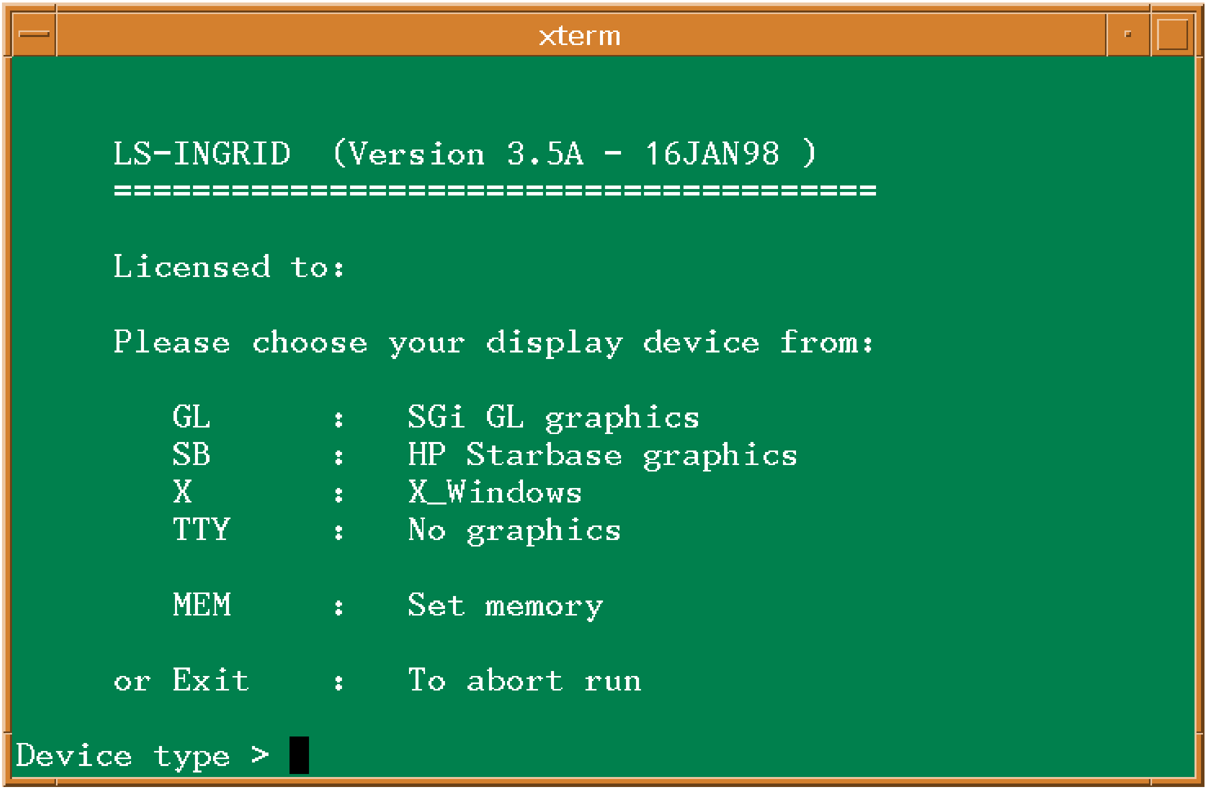
```
ingrid i=test.ing o=test.out gl
```

To define both input and output filenames, and to use the Silicon Graphics GL graphics protocol. However it is also quite acceptable to have no command line options:
In this case you will be prompted for all the input required and filenames will revert to their

defaults.

2.2 Interactive graphics device definition

If a graphics device has not been given on the command line then you will be prompted for one:



```
     LS-INGRID  (Version 3.5A - 16JAN98 )
     =======================================

     Licensed to:

     Please choose your display device from:

        GL       :   SGi GL graphics
        SB       :   HP Starbase graphics
        X        :   X_Windows
        TTY      :   No graphics

        MEM      :   Set memory

     or Exit     :   To abort run

Device type >
```

3.0 THE GRAPHICAL USER INTERFACE

3.1 Description

Figure 1 shows a typical screen from LS-INGRID. It is divided into several areas by function as follows:

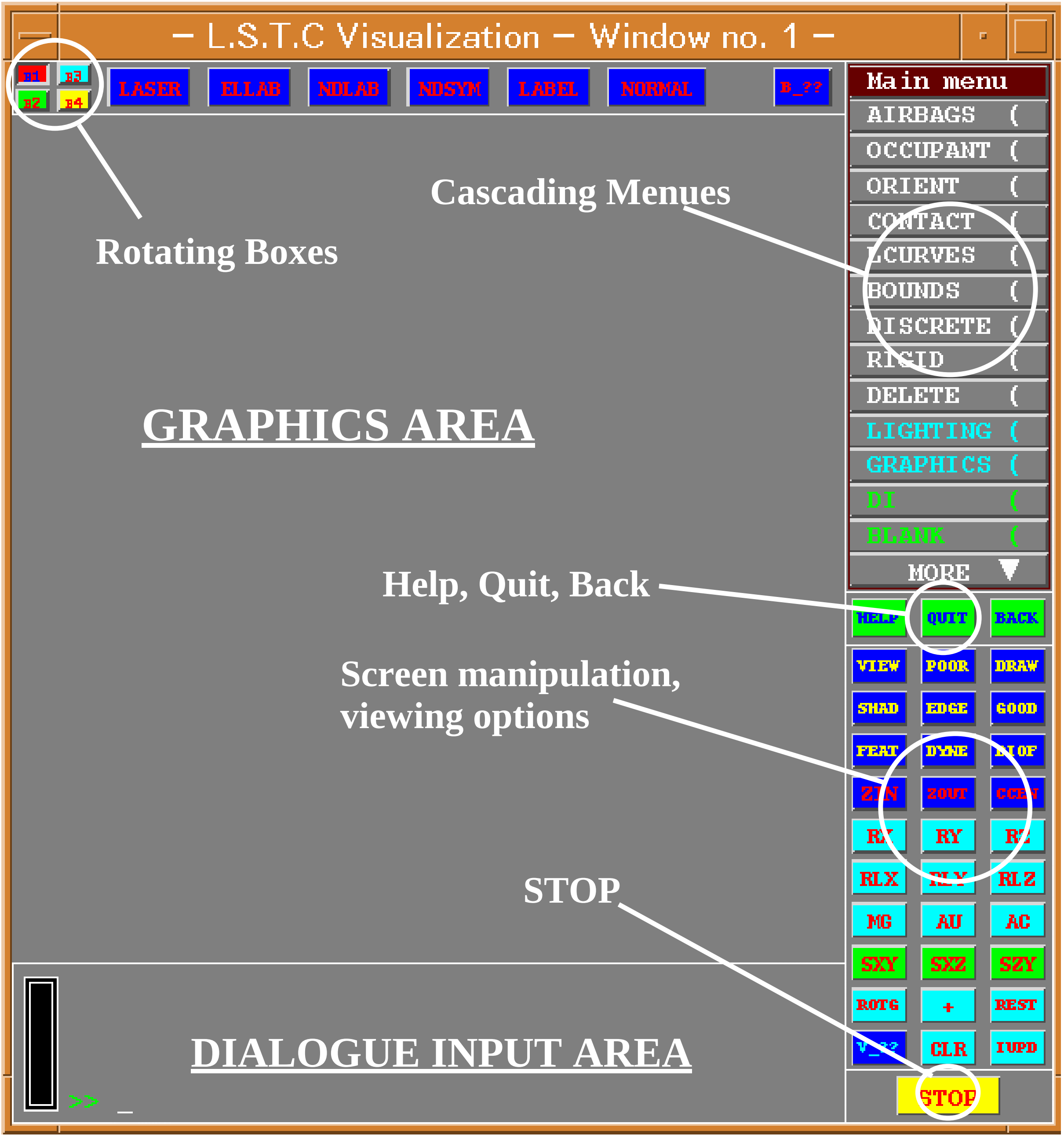


Figure 1: Typical LS-INGRID screen interface layout

GRAPHICS AREA: All graphics appears in here, and it is also used for listing large quantities of text. Graphical output is clipped to the area borders.

DIALOGUE AREA: The user may type in commands and receive text output from the program in this area. Generally short messages are written here whereas longer ones are directed to the graphics area. A scroll bar is provided at the left hand side of this area so that the last 60 lines of text may be scrolled up and down for review.

CASCADING MENUS: The program always has at least one command "menu" live on the screen and, depending upon where the user is in the command tree, there may be several overlaid in this area. The most recent menu is the active one, but its predecessors are shown in order to remind the user of where he is in the program. Menu selections are made with the mouse by clicking on the appropriate bar.

VIEW MANIPULATION: This box holds commands that control the current view; for example rotation, scaling, translation, display mode; these are described fully in section 6.1. On-line help on commands in this box may be obtained using the **V_??** button in it. Users on 3D devices have further "dynamic viewing" options that are described in section 3.4.

ROTATING BUTTONS: Across the top of the screen is a rotating box of four sets of button. (Consider the number plate on James Bond's car.)

These are called B1 through to B4, and the first two are set aside for common functions, the last pair are reserved for special functions related to the current operation. On-line help on each set may be obtained by using the **B_??** button in that box.

HELP, QUIT, BACK: These buttons are always globally active, and can be used anywhere to ask for "help", to "quit" (abort) the current command, and go one level "back" up the menu tree.

STOP: Finally this button will "stop" execution immediately. Use with care!

All menu and button options are selected with the left mouse button. The other two buttons are not used in the graphical user interface, (although they are used for dynamic viewing: see section 3.4). Selection is made when the mouse button is released therefore it may be held down and "dragged" over items until a selection is made.

3.2 Customizing the user interface

The user has no control over the layout or contents of the various screen areas, but he may control the size of the text used for dialogue and menus. This is done as follows:

- File ".ingdat" is created containing the size in cm for the text in each of these areas. For example it might read:

  ```
  menu_size:    0.55
  dialog_size:  0.40
  ```

  The text should be in lower case, and the sizes may be entered in free format.

- This file is then placed in:

  either: The current working directory;
  or: The users home directory;
  or: The $LSTC directory.

  They are searched in this order, so a file found in the current directory will be used in preference to one in the home directory, and so on.

- The program <u>must be restarted</u> for these settings to have an effect.

Note that using very large text may make the menus hard to use as they will be reduced to only a few column's width, and the number of rows that can be displayed will reduce. A reasonable value is usually about 0.4cm, corresponding to a 12 point font.

Likewise the number of rows of text available in the dialogue area will depend on the font size used, and there is a similar trade-off between legibility and amount of information that can be displayed.

<u>Relationship between text height requested and fonts used.</u>

Fonts tend not to be infinitely variable in size, but rather to come in discrete sizes (measured in "points"). Typically LS-INGRID will use 8, 10, 12, 14, 18 and 24 point fonts. All text sizes are translated into the nearest fixed size font available on the machine, so small changes in requested size may have no effect until they "jump up" to the next font.

3.3 Interaction between user interface and display hardware

Generally LS-INGRID will attempt to present a consistent appearance over a range of graphics hardware and protocols. However there are some minima:

- The display should have at least four bit-planes if single-buffered, or 8 bit-planes (i.e. 4:4) if double-buffered. This is because it needs at least 16 colors to function well, and #colors = $2^{\#bit\text{-}planes}$. On displays with fewer planes the colors will default back to black or white as appropriate.

- If shaded and lit images are required then the display should have at least 8 bit-planes (single buffered), or 16 arranged as 8:8 (double-buffered). In the absence of this shaded images may not be rendered at all, or may come out as solid primary colors.

X_Windows users:

- Should use at least a four bit-plane PseudoColor visual, and an 8 bit-plane would be better. 4, 8, 12 and 24 bit-plane depths are supported as PseudoColor, DirectColor and TrueColor visuals.

SGi GL users:

- Need not worry about this. All depths of screen are supported.

HP Starbase users:

- Should ideally use a 24 bit-plane TrueColor visual double-buffered to 12:12 bit-planes. However 8:8 and 4:4 double-buffered True and PseudoColor visuals are also supported, although the latter will not provide shaded images.

TTY (No graphics) users:

- This also does not use the screen interface, and LS-INGRID reverts to batch mode.

3.4 Dynamic viewing on 3D devices

On SGi GL, HP Starbase and on X_Windows devices LS-INGRID supports "dynamic viewing". This means that the image on these machines can be rotated, scaled and translated in real-time using the mouse. This is not supported on all 2D graphics protocols.

The commands to do this are:

VIEWING FUNCTION	KEY COMBINATION	CURSOR SYMBOL
XY ROTATION Z ROTATION	***<left shift>+<left mouse>*** " " " "	**[XY]** (red) **[Z]** (red)
XY TRANSLATION	***<left shift>+<middle mouse>***	**[+]** (yellow)
SCALING	***<left shift>+<right mouse>***	**[=]** (green)

Dynamic viewing commands.

Notes: Rotation is always about the screen axes. XY rotation being selected if the mouse starts in the middle 2/3rds of the screen, Z rotation if it starts outside this area. Rotation tracks mouse motion assuming an origin at the center of the screen.

Translations are always in the screen XY plane, and the model will track the mouse movements. Initial mouse position does not matter.

Scaling uses motion up and to the right will make the image bigger, down and to the left smaller. Initial mouse position does not matter.

The cursor symbols shown in this table are approximate: the actual symbols used are clearly recognizable.

"Wireframe" mode option for faster viewing

The dynamic viewing keys combinations described above will operate on the image in its current mode, for example hidden-line or shaded. On some hardware this can be slow if the image is big, so there is an alternative of performing the dynamic viewing operation in temporary "wireframe" mode. This is invoked by using the ***<left control>*** key with the appropriate mouse button, instead of the ***<left shift>*** one above.

If this is done the image is transformed using only vectors and text: all hidden-surface removal and shading is suppressed for the duration of the operation. Once the ***<left control>*** key is released the image will be redrawn in its original mode.

Interaction with explicit viewing commands

Finally: dynamic viewing is complementary to the explicit viewing commands in the lower right hand button box. The two methods may be used and interchanged freely as they both update the current view transformation matrix.

3.5 Laser hard-copies

LS-INGRID can make laser prints in two ways, both of them invoked by the

LASER

option in the top button box. The two modes are:

Raster dumps: May be made of the whole window in Postscript, HPGL and other formats. These are complete dumps of the whole screen, menus and all, written pixel by pixel to the laser file, therefore the files can be large. However they can capture complete images on 3D devices.

Vector dumps: These are dumps of the image only, not the user interface, and are generated as vectors from the "redraw stack" of the graphics driver. They only contain 2D vectors, text and polygons, so on 3D devices most of the image will not appear. Therefore only use these on 2D X_windows devices.

In either mode laser files are written using the following name convention:

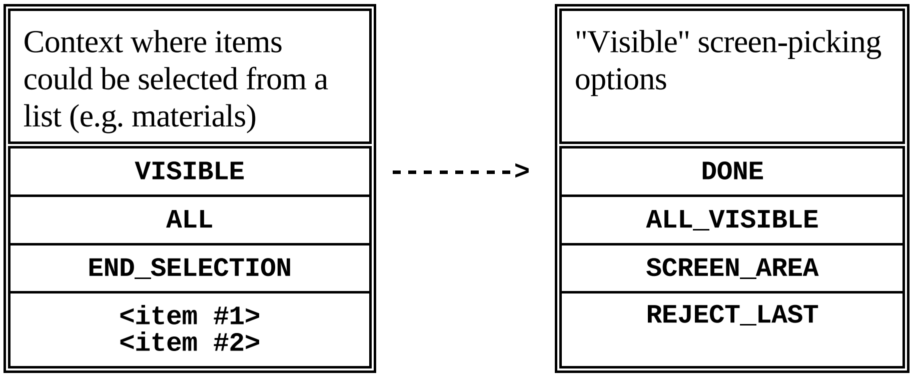
Postscript format:	**dig1.ps, dig2.ps, etc**
HP Laserjet format:	**dig1.pcl, dig2.pcl, etc**
HP Paintjet format:	**dig1.pj, dig2.pj, etc**
Colorjet format:	**dig1.cj, dig2.cj, etc**
Deskjet format:	**dig1.dj, dig2,dj, etc**

Note also that the standard X11 utilities ***xwd*** and ***xpr*** can also be used to capture screen images and format them for laser printer output. Refer to the Unix manual pages for more information.

3.6 "Screen picking": selecting visible items using the cursor/mouse

In an interactive, graphical program you often want to select a visible item from the screen without having to know exactly what its label of identifier is.

This concept is referred to as "screen-picking", and LS-INGRID has a standard set of screen-picking options that appear in most contexts where you are asked to select entities. These appear in the menu structure as follows:



Context where items could be selected from a list (e.g. materials)

VISIBLE

ALL

END_SELECTION

<item #1>
<item #2>

-------->

"Visible" screen-picking options

DONE

ALL_VISIBLE

SCREEN_AREA

REJECT_LAST

By selecting ***VISIBLE*** in the left hand menu you will pass to the "visible options" one on the right which handles screen picks. The options here are:

<pick an item> Either node location or element center depending on context. Materials, surfaces and parts may be selected by picking any relevant element on them.

<type a label> You can type in the entity labels explicitly

DONE To terminate picking selection. The contents of the current cursor pick list will be added to anything selected explicitly, and control will pass back to the menu on the left for further explicit selections.

ALL_VISIBLE Selects all visible items of the relevant type. "All visible" means all that have been drawn on the screen, i.e. not those, which are blanked, or otherwise removed from the display.

SCREEN_AREA To select all items within a screen rectangle defined by picking its opposite corners with the mouse.

Finally, when you have returned to the menu on the left, you can terminate data entry with ***END_SELECTION***. This will return with all the items selected, whether explicitly or by picking, and continue processing.

Cases where only the "screen-picking" menu appears

When picking explicit nodes or elements it would be impossible to list all available node/element labels in the menu system: there would be too many of them.

In this case only the right hand, "screen-picking", menu is displayed and usage continues as before. ***DONE*** terminates data entry and control returns immediately with the items picked.

Cases where only one item is to be screen-picked

When only one item is required then the screen-picking menu can be dispensed with altogether. In this case simply pick the entity directly, and control will return as soon as one pick has been made.

Errors during screen-picking

- The message "Nothing close enough" appears:

  This means that the cursor position was too far away from any relevant item. Try again being more precise. Note that picking elements, materials, contact segments, parts, etc requires you to pick the center of an element. You don't have to locate it exactly, but you should try to be reasonably close.

- You seem to have selected the wrong entity:

  This can be a problem when picking from hidden-surface plots on 3D devices: LS-INGRID cannot know exactly what has been displayed and obscured by the hardware Z buffer. It tries to pick the item nearest to the observer if an ambiguity occurs, but occasionally this gives the wrong result. Use ***REJECT_LAST*** to cancel the last pick, and try again, maybe after having adjusted the view a bit.

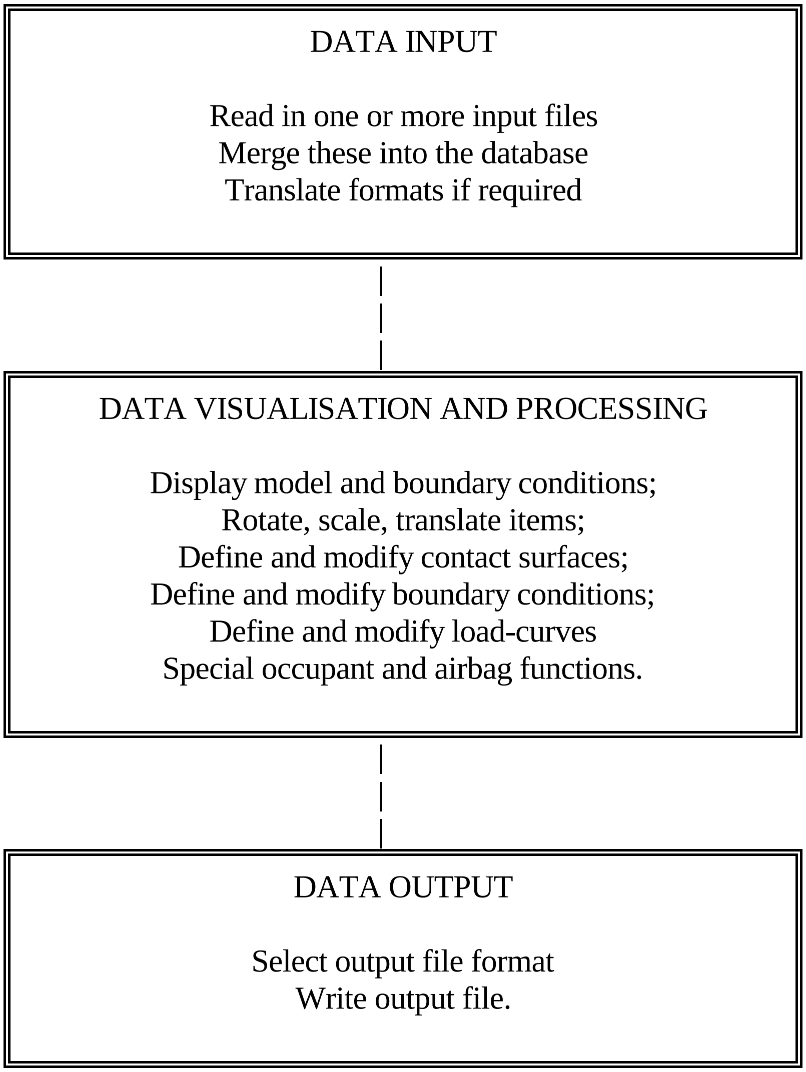
- You don't seem to be able to pick a visible entity:

  If you have ***SKETCH***ed something, or superimposed it on a plot using a ***DI***splay Option, you will not be able to pick it. This is because displaying it in this way does not flag it as "visible" in the program's internal tables.

4.0 USING LS-INGRID

4.1 Control flow through the program

LS-INGRID operates in three distinct, sequential stages:



DATA INPUT

Read in one or more input files
Merge these into the database
Translate formats if required

DATA VISUALISATION AND PROCESSING

Display model and boundary conditions;
Rotate, scale, translate items;
Define and modify contact surfaces;
Define and modify boundary conditions;
Define and modify load-curves
Special occupant and airbag functions.

DATA OUTPUT

Select output file format
Write output file.

At the present stage of development of the program it is not possible to reverse the order of these operations. For example it is not possible to decide in the middle of the processing stage to read in another input deck, nor is it possible to write out an intermediate set of results and then to continue processing.

Users of LS-INGRID in "batch" mode, and those using the ***TTY*** device, will tend not to use the middle data processing part.

4.2 Data input

If "batch" mode (see 2.1) is not used you will see the main data input screen once you have selected an input device:

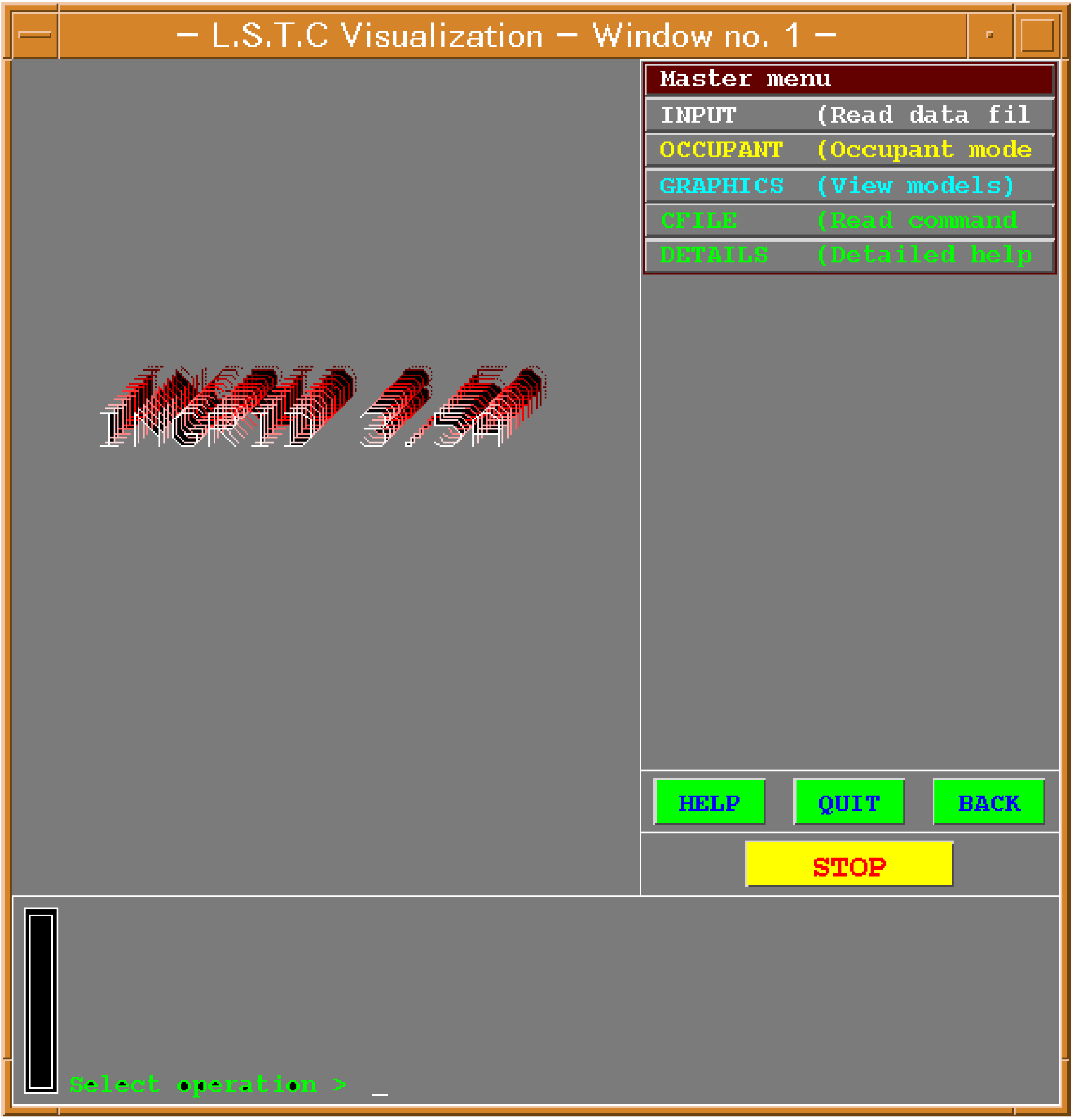


Figure 2: Data entry main screen.

4.2.1 **INPUT** Read in input files

Use this option to read files into LS-INGRID. Supported formats are:

INGRID Traditional "ingrid" format file. Do not attempt to read in more than one of these and, if used in conjunction with other file formats, read this one first.

KEYWORD Reads LS-DYNA3D 930 "keyword" format input files.
DYNA3D Reads all LS-DYNA3D "fixed" format input files.
VECDYNA Reads LS-VECDYNA input files.
MVMA Reads MVMA-DYNA3D input files.
OASYS Reads OASYS DYNA3D level 5 input files.

NIKE3D Reads LS-NIKE3D input files (not operative yet).
NASTRAN Reads NASTRAN "bulk data" input files.
IDEAS Reads SDRC I_DEAS "universal" files.
PATRAN Reads PDA PATRAN "neutral" files.

Any number of files, other than "ingrid" format ones, may be read in any order. Data will be extracted from them and stored in the database. The data in each file will become the next highest "part" in LS-INGRID, so while the models are combined it is still possible to operate on each one separately by selecting its "part" id.

"Ingrid" format files may generate many parts, so the above statement is still true but note that files read in afterwards may generate higher part numbers than you expect.

Handling of Node, Element and Material numbers read from multiple files

If you read more than one model in multiple files it will be necessary to provide an offset for the node, element and material labels so that the numbering systems do not overlap internally.

You are prompted for this automatically, and given the following options in a
"Label offsets" menu:

`No gaps` (Default). This is a number that will leave no gaps in the external numbering system.

`Small offset` This will give a small offset, set to the next sensible power of 10.

`Large offset` Will give a larger offset to the next higher power of 10 above the "small" one.

`No offsets` No offset at all. Use this with great care, and only if you are certain that there is no overlap between the numbering systems in models.

Handling of other item numbers

All other numbered items from multiple files; for example contact surfaces, load-curves, etc; will be renumbered to sequential order with no gaps.

Processing of items in DYNA3D format files

All items in DYNA3D files are processed and stored by LS-INGRID except:

- Traction loads on beams;
- Momentum deposition on solids;
- Tied nodes with failure;
- Nodes moved by an interface file;
- User supplied interface data;
- Superplastic analysis options;
- Implicit springback options;
- Rigid/deformable material switching.

In addition users should note that some variants of DYNA3D format contain concepts that are not translatable to others: for example some contact options. In these cases the end result will tend to be to reset these parameters to their defaults.

Processing of non-DYNA3D formats

When "Nastran", "Ideas" or "Patran" format files are read in only a limited subset of data can be extracted. Generally you may expect to import the following:

Nodes	Label and coordinates.
Elements	Compatible element topologies.
Materials	Treated as type 1 (Elastic), using Youngs Modulus (E) and density in most formats. Some formats also handle type 20 (Rigid) and type 24 (Elasto-plastic) from the relevant material data.
Properties	Shell thicknesses from some formats.
Loads	Nodal forces and element pressures from some formats.
Restraints	Nodal restraints (SPCs) from some formats.
MPCs	(Multi-point constraints) from NASTRAN only.
Gravity	(As acceleration) from NASTRAN only

You can always expect to read geometry, but treat any extra information as an added bonus !!

4.2.2 `OCCUPANT` : Read in occupant (dummy) models

Up to 5 pre-existing occupant models may be read at a time into LS-INGRID. Such models are normal input decks, in any of the formats described above, but they have the option of having a "tree" file associated with them. This "tree" provides extra information about the connectivity of the model components, and permits LS-INGRID to move limbs intelligently in the visualisation part of the program.

The structure of a "tree" file is described in Appendix I.

In all other respects occupant models will be read in, incremented and processed in exactly the same way as other input files.

For a model to appear in this menu it must be notified to LS-INGRID in the ".ingdat" file described in section 2.2. Any number of occupant models may be described in this file by adding lines as follows:

```
dummy_spec:  <acronym>  <file type>  <pathname for file>
```

Where:

dummy_spec: Is the special command stating that the line contains dummy information.

<acronym> Character string (uppercase and up to 10 chars) by which the dummy will be known within the program.

<file type> The input file type in lower case, i.e. "Dyna3d", "Nastran".

<pathname> Is the directory in which the files are to be found.

Therefore a typical ".ingdat" file might contain the lines:

```
dummy_spec:   RH3      oasys    /users/occupants/rh3
dummy_spec:   SID      dyna3d   /users/occupants/sidney
```

Within the directory specified LS-INGRID will expect to find the following files, all names (including ***<acronym>***) being lower case:

`<acronym>.inf` Input deck of model. This is required.

`<acronym>.tree` "Tree" file for dummy. This is optional.

`<acronym>.list` Text file giving brief description of the dummy. This is optional.

For example the ".ingdat" file entries above would make the dummy models available in the ***OCCUPANT*** menu:

```
RH3
SID
```

And if you selected the ***RH3*** option LS-INGRID would expect to find files:

```
/users/occupants/rh3/rh3.inf
/users/occupants/rh3/rh3.tree
/users/occupants/rh3/rh3.list
```

If the "tree" file is missing

LS-INGRID can still process and add seat-belts to the occupant, but it will not be able to move its component parts in a connected fashion.

If the "list" file is missing

LS-INGRID's function is unaffected, but the ***LIST*** option in the **OCCUPANT** menu will not be able to tell the user anything about the dummy.

Limits of occupant functions

There may be any number of occupants defined in the ".ingdat" file, of which up to 5 may be read into LS-INGRID concurrently. There are some limits to what may be defined in "tree" files: see Appendix I for details.

To use the same occupant model more than once

Read the same file in as many times as required. You cannot duplicate models by copying in the visualisation part of LS-INGRID.

4.2.3 GRAPHICS : Enter the visualisation part of the Program

When you have read in everything you want to, (remember that you cannot go back for more), use this command to enter the visualisation section.

There will be a slight pause when this happens as LS-INGRID sorts through what it has read in. A summary of all contact data and all materials will be printed to the original text window, and then the screen will be cleared and the main graphics interface will appear as described in section 3.1.

4.3 The Data Visualisation and Processing Menu

Once you enter section 2, data processing and visualisation, the "Main Menu" will be mapped. This may be thought of as the top of a command tree, below which there are levels of sub-menus. You cannot move higher up the tree than this level.

Although there is only one "Main Menu" its functions are split between data processing and data visualisation.

4.3.1 The main data processing options in LS-INGRID 3.5A

`AIRBAGS` Controls airbag folding. Folds can be created, modified and deleted; the mesh can be folded interactively and the effects visualized.

`OCCUPANT` Controls occupant (dummy) positioning, seat-belt tensioning, and the definition of related entities: slip-rings, retractors, sensors, pre-tensioners and accelerometers.

`ORIENT` Rotates, translates, reflects and scales the whole model, selected parts, selected materials or individual nodes and elements.

`CONTACT` Contact surfaces may be created, modified and deleted. Their parameters may be listed and viewed.

`LCURVES` Load-curves may be created, modified and deleted. Any or all curves may be listed and plotted.

`BOUNDS` Boundary conditions may be applied to the model: initial velocities, restraints, applied velocities/accelerations/displacements and base accelerations may be applied to nodes and rigid bodies. All of these may be created, modified, deleted, listed and viewed.

`SPRING` Spring element processing. At present springs may only be listed.

The use of these commands is described fully in section 5 of this manual.

4.3.2 The main data visualisation options in LS-INGRID 3.5A

LIGHTING Controls the lighting parameters used for shaded plots. Up to ten light sources + ambient light may be defined, and surface visual properties set.

GRAPHICS Controls parameters of the graphics display: element shrink percentage, hidden-line resolution, etc.

DI "DIsplay options". Permits many different types of data to be overlaid on the basic plots.

BLANK May be used to add/remove entities from the display list, and so speed up and clarify graphics.

The use of these commands is described fully in section 6 of this manual.

4.4 Data Output

The last two commands in the data processing and visualisation main menu control and execute output. They are:

OUTPUT Allows selection of output file format (e.g. DYNA3D, Keyword, etc)

CONTINUE Moves execution to the 3rd, output stage of LS-INGRID.

Remember that once execution has passed to the output stage of LS-INGRID you cannot return to either of the two previous stages.

In particular this means that you cannot write an output file of your current model, and then continue to process it.
Nor can you write more than one file of output during an LS-INGRID run, i.e., you cannot directly write out the same data in two or more different formats.

4.4.1 `OUTPUT` : Selection of output format

Unless LS-INGRID has been run in batch mode it does not know what analysis code format it should use for its output (**`ingrido`**) file. This command allows you to select a target format from the options:

`DN3D` DYNA3D format. The variants available cover all contemporary and past LS-DYNA3D formats plus some alternatives. The author recommends that the **`V93`** (i.e. LS-930, formatted) option be used.

`DN2D` LS-DYNA2D format. The user must choose which of the LS-INGRID [X,Y,Z] coordinates are mapped onto the DYNA2D [R,Z] axes.

`NK3D` LS-NIKE3D implicit structural analysis code format.

`NK2D` LS-NIKE2D format. As with DYNA2D the [X,Y,Z] to [R,Z] axis mapping must be defined.

`TZ3D` LS-TOPAZ3D thermal analysis code format.

`TZ2D` LS-TOPAZ2D format. [X,Y,Z] to [R,Z] must be defined.

`FACE` FACET view factor calculation code format.

`RROU` "Rolls Royce output": proprietary Rolls Royce format.

`PNEU` PDA PATRAN neutral file format.

4.4.2 `CONTINUE` : Finish visualisation and generate output

If no output format has been defined the user is forced to select one from the options above; then the visualisation part of the program is terminated, the user interface is shut down, the output file is generated and the program exits.

Output is to filename **`ingrido`** unless the ***`<o=filename>`*** command-line option was used: see section 2.1.

5.0 USING DATA PROCESSING MAIN MENU COMMANDS

This section describes how to use the various main menu commands listed in section 4.3.1 above. This is a summary guide, and reference should be made to the on-line "help" available in each set of sub-menus. A description of the visualisation commands may be found in section 6.0.

Summary of on-line help and command abort options

Users are reminded that on-line help is always available in LS-INGRID in the following forms:

`HELP` button — Is **always** globally active. It may always be used to request help and advice whenever input is required. Generally it will give a descriptive text, a list of commands available, and some guidance.

`EXPLAIN` — This menu option is available at many prompts where the "help" text alone would not be adequate. It will give more information about the current activity.

`DETAILS` — In a few particularly complex areas this menu option will be available. It is used to select "detailed" help on a particular command: first select DETAILS, then the <command> in question.

In addition to these there are two special "help" options:

`V_??` button — In the viewing control box (on the lower right hand side). This button acts like the "details" command above: first click on ***`V_??`***, then on the ***`<viewing command>`*** in that box on which you want information.

`B_??` button — In the rotating boxes across the top of the screen. This is used the same way: ***`B_??, <box command>`*** will explain what that command does. This is particularly useful in boxes B3 and B4 where commands will change with context.

Finally, to abort commands if things go wrong:

`QUIT` button — **Always** globally active, and may always be used whenever input is required. It will terminate the current operation and return control to the next highest command prompt.

5.1 AIRBAGS: Airbag folding options

Summary: Used solely for folding airbags. Up to 50 folds may be defined on an existing mesh. Folds can be created, modified, deleted and visualised. Fold definitions may be stored for later re-use.

Required: A shell element mesh, topologically flat, lying in the X-Y global plane.

Airbags are enclosed volumes of shell element meshes, generally using the special airbag material number 34 that will be inflated during the analysis using "control volumes". It is very hard to mesh these in their initial folded state, so instead they are meshed flat and then folded in LS-INGRID.

5.1.1 Initial steps to go through

(a) Mesh the bag, either externally or within LS-INGRID, with a view to folding it. It is best if there are mesh lines on or near the intended fold lines. The unfolded mesh should be topologically flat, or at least made up of flat layers stacked above one another, and it must lie in the global XY plane.

(b) Use the ***AIRBAGS*** main menu command to enter the airbag folding menu You will note that rotating button box B4 becomes active across the top of the screen with special folding options.

(c) Use the ***[AIRBAGS] SELECT*** command to define, which parts of your mesh are to be folded. This is done by setting a flag against every node in the model stating whether or not it is to be included for folding. By default **all** parts of the model are pre-selected for folding, but you can reduce this to a subset by ***ADD*** or ***SUBTRACT*** commands as required. You can see which nodes have been selected by using the ***SKETCH*** command: this circles every node on the current image selected for folding.

At this stage you have sorted out what is going to be folded, and you can now start creating folds.

Hints to help you:

- If you have a large model you may find it helpful to ***BLANK*** everything except the airbag at this stage. It will make it easier to see what is going on.
- If your airbag does not lie in the XY global plane you will have to ***ORIENT*** it until it does before you commence folding. Once the bag has been folded you can move it back to its correct orientation and position in the parent model.

5.1.2 Creating a fold in the airbag

In the ***AIRBAGS*** menu go through the following steps:

(d) Use the ***DEFINE*** command to create folds. First you have to define whether the fold is "thin", "thick", "spiral" or "tuck". A thin fold is like the crease when you fold paper: it has no radius; a thick fold instead has a definite radius. Spiral fold is like rolling paper, therefore it requires a clearance and the inner radius. A tuck fold will tuck the material to be folded in between the top and the bottom layers.

THIN	Choose a "thin" fold;
THICK	Choose a "thick" fold;
SPIRAL	Choose a "spiral" fold;
TUCK	Choose a "tuck" fold.

(e) Now you have to define the fold itself, i.e. to define the location of the fold line.
A fold should lie perpendicular to either the X or the Y axis and may be created by:

NODE_NODE	To make it line between two nodes (screen picked);
COORDINATE	To define an explicit axis and a distance along it; or
3D_LINE	Select a predefined 3D-line (not yet implemented)

Once the fold line has been defined it will be checked. If it does not lie in the X or Y axes you will be forced to redefine it, or to "snap" it onto the nearer axis.

(f) You must decide on which side of the fold-line the material is to be moved by selecting one of (applicable only in ***COORDINATE***):

+ or ***-***	For the positive or negative side of the axis;
PICK	To select the side by picking any node on it.

(g) You define whether the folded portion is to move "up" over the unfolded part, or "down" underneath it:

UP	Folds it up and over;
DOWN	Folds it down and under.

(h) Then you must define the fold dimensions (see Appendix). These are:

THICKNESS This is the characteristic separation distance between layers, and a good initial value is the material thickness: typically of the order of 1mm.

SCALE FACTOR The scale factor used to expand/contract the fold horizontally. Start with a value of 1.0.

FOLD RADIUS Only applicable to "thick" folds. Use the material thickness.

(i) The fold line will then be sketched on the mesh and you will be asked to confirm whether it acceptable or not:

OK Fold is acceptable, so store it;
NEW Fold is not acceptable, so re-define it;

To abort the operation here, or at any stage above, use ***QUIT***.

(j) The fold is now defined, but the mesh has not actually been folded yet. To make a fold happen you must use the command:

EXECUTE To execute the current fold(s) on the selected nodes.

The options under this command are:

0 To seal edges only. This is only applicable to some meshes defined in "batch mode" within LS-INGRID where the layers are not connected. Externally generated meshes can ignore this command.

ALL To execute all current folds.

1 To execute fold #1 only;
2 To execute folds #1 to #2 only;
3 To execute folds #1 to #3 only;
etc To the highest fold number defined.

Initially select ***ALL*** so that all folds are executed, and define the minimum thickness to be used for each fold. For each fold the greater of this thickness and that defined for the fold is used, so a good starting point is again the material thickness.

You have now created a fold: to see it redraw the mesh in any display mode. You can go back and repeat steps (d) to (j) above to create more folds, the maximum number you may have being 50. Note that:

- Folds are always created in ascending, sequential order.
- Folds are always executed in the same ascending, sequential order.
- Intermediate folds may punch parts of the bag "through itself" if desired, so long as later folds undo this situation to leave a bag without self-penetrations.
- Ones first attempts at folding often go wrong! Don't despair, and read the next sections to see what to do about it.

5.1.3 Folding diagnostic and checking options

The following ***AIRBAG*** menu options will help you with creating folded bags:

CHECK Checks that the geometry of fold definitions is OK. In particular it checks that "thin" folds lie exactly on mesh lines since, if they do not, the fold geometry can go subtly wrong. Any nodes lying close enough to a fold to be on it, but not exactly on it, are circled on the plot and you are given the option of "snapping" them exactly onto the fold line:

OK	"Snap" the offending nodes onto their respective fold lines;
LEAVE	Leave them alone. You may have to modify the folds;
CH_TOL	Change the checking tolerance if too many nodes flagged.

STEP_BY Is another useful diagnostic tool. If turned on the model will be re-drawn after each fold is carried out so that you see an animation of the fold sequence as it happens. You may give a delay (in seconds) between frames.

RESET "Undoes" all folds by resetting nodes back to their unfolded geometry. The fold definitions remain, and the airbag can be refolded at any time.

LIST Lists the parameters of the selected fold to the dialogue area.

SKETCH Sketches the selected fold lines, numbers and key data on the current mesh. (See also "global fold options" below.)

SUMMARY List the parameters of **all** folds in the screen graphics area. This destroys the current image so use ***LIST*** instead if you don't want this to happen.

5.1.4 Saving fold definitions on disk

It is possible to save all current fold definitions in a file. This is useful if you are about to embark on modifications and want a fallback position. Also the saved definitions are model independent and may be applied to other, similar meshes.

SAVE	Saves all current fold definitions to a file;
RELOAD	Reloads saved definitions, overwriting any current ones.

5.1.5 Special "Global" folding options

You will notice that when you enter the ***AIRBAGS*** menu button box B4 across the top of the screen becomes active with special "global" folding options. These are:

SKFALL	Sketch all folds. Shorthand for ***SKETCH ALL*** in the ***AIRBAG*** menu.
SKXFOL	Sketch all folds perpendicular to the X axis.
SKYFOL	Sketch all folds perpendicular to the Y axis.
REFOLD	Re-executes all folds selected by the ***EXECUTE*** command.

They are provided for convenience since they are commonly used options.

5.1.6 Modifying existing folds

You can change any parameter of any fold with the ***MODIFY*** command. To use it:

(a) Select the fold to modify from the list provided. Only one fold may be modified at a time. Note that you are always working on a "scratch" copy of the fold, and changes made have to be saved (see below) to make them permanent. So mistakes don't matter.

(b) Select a parameter to modify from the options:

V1 Unfolded position. This is the location of the fold line on the **unfolded** mesh geometry. You will be provided with the options:

<coord>	Current fold coordinate (in X or Y as appropriate)
NODE	Uses the relevant coordinate of a picked node;
AS_FO	Uses the current **folded** position (see below).

V2 Folded position. This is the location at which the fold will take place: usually the same as the unfolded position, but not always. Options are:

<coord>	Current fold coordinate (in X or Y as appropriate)
NODE	Uses the relevant coordinate of a picked node;
AS_UN	Uses the current **unfolded** position (see above).

The "hints on folding" below give and explanation and more advice on the subject of folded and unfolded coordinates.

V3 Selects whether to fold mesh on the +ve or -ve side of the fold line.

V4 Lets you change the fold thickness.

V5 Selects whether to fold "up" over the mesh, or "down" under it.

V6 Changes the fold direction between global X and Y.

V7 Changes fold type between "thin" and "thick" logic.

V8 Changes the fold scale factor.

V9 Changes the fold radius.

Non-expert users should not attempt to use the ***FOPT, FIGN*** and ***FMOV*** options in this menu: they are very specialist.

(c) View the effects of the changes made.

You will observe that using the ***MODIFY*** option activates rotating button box B3 across the top of the screen. These options are intended to make fold modification easier by providing ways of seeing the effects of changes:

CLIST Lists the current (changed) settings of the fold to the dialogue area;

OLIST Lists the original parameters of the fold to the dialogue area;

RESET Resets the current parameters to their original (unchanged) state;

SKETCH Sketches the current fold on the mesh as currently drawn;

UPDATE Is a toggle flag which, if set, will cause the mesh to be re-drawn every time a fold parameter is changed. This is very powerful since it shows immediately the effect of changes.

(d) Decide whether or not to save the fold changes.

Since ***MODIFY*** always works on a "scratch" copy of the fold you must choose explicitly to save your changes:

SAVE Copies changes to the permanent fold definition and exits;

QUIT Exits leaving the permanent fold definition unchanged.

5.1.7 Deleting, copying and reordering folds

(a) Deleting folds

This is simple. Use the ***DELETE*** option and select those folds you wish to delete. The remaining folds will be renumbered if necessary so that they start at #1 and rise sequentially with no gaps.

(b) Copying folds

Use the ***COPY*** option and select the fold(s) to be copied. (Folds are prefixed "OLD_FOLD_n" to make the point that they are existing ones.)

COINCIDENT Copies the fold(s) exactly with no changes to location. You will have to use ***MODIFY*** subsequently to change the location(s).

REFLECT Copies the fold(s), but reflects coordinates about the point given, and reverses the +ve/-ve flag. Since many fold patterns are symmetric about a bag centerline this can be very useful since only half the folds need to be defined explicitly.

(c) Reordering folds

Folds are **always** applied in the order in which they are defined, so to change the order in which they occur a fold must be moved from an "old" number to a "new" one.

The ***REORDER*** option lets you pick an "Old" fold number and its "New" number. The intervening folds are shuffled up or down as necessary.

5.1.8 Hints on folding

(a) Plan your fold pattern in advance, and try to organise the mesh so that straight mesh lines are available in the right places for folding.

(b) Building folds with the correct material thickness can sometimes be difficult since the folds are too thin to be seen easily. Try using the ***EXECUTE*** command with an artificially large value when creating folds, then correct it later. Note that excessive mesh distortions occur if too great a thickness is used. See also (d) below.

(c) Try to organize folds so that they are defined from the outside of the mesh inwards. This is not mandatory, but it means that a fold's "unfolded" and "folded" positions are the same. Otherwise folds must be defined on geometry that has itself been changed by preceding folds: the "unfolded" position is simple (it is always the fold location on the unfolded mesh), but the "folded" position (where the fold line will lie after previous folds have moved it) must be computed by the user.

(d) "Thin" folds are better than "thick" ones since they don't require very small elements around the fold radius, but they do distort the mesh. Therefore a good solution is:

- When creating folds make each individual fold thickness artificially small. The author tends to use 10% of the actual material thickness.

- When using the ***EXECUTE*** command during fold definition make the over-riding thickness something sensible: i.e. the real material thickness. Since the greater of this value and the individual fold thickness values are used the resulting fold pattern will be thick enough to be visualized. (Very thin folds are impossible to see!)

- When all folds have been defined and checked re-issue the ***EXECUTE*** command with an artificially small value too. This will result in the folded mesh being very compressed, and so element distortions around thin folds will be minimal.

- When defining the special airbag contact (type a13) for the bag use the "loadcurve controlling thickness" option to ramp up the thickness used for contact from this very small value at time zero to a realistic value at (say) 2mS. This will prevent contact initialisation from detecting the initial penetrations that do in fact exist at time zero.

(f) If you get initial penetrations during contact initialisation in DYNA3D you may be able to use the loadcurve-defined thickness described above to fix them.
But you should also check your mesh visually for errors. Do this by:

- Aligning the view so that you are looking exactly down the X or Y axis as appropriate.
(Use the ***SXZ*** or ***SXY*** commands in the Viewing button box.)

- Zoom in on the folds looking for crossed elements, or elements too close to one another (especially at corners).

Initial penetrations often occur in higher numbered thin folds when many layers are being folded. Try modifying the **SCALE** factor to space out layers.

5.1.9 Worked examples

The **`/examples`** directory supplied with LS-INGRID contains some examples of airbag meshes and fold definitions:

`driver.ing` Is a circular ("pancake") driver's side bag meshed in a way that makes folding easy.

`pillow.ing` Is a squarer ("pillow") passenger side bag, again meshed in a way that makes folding easy.

Appendix II gives examples of how to fold and deploy these bags.

Permission is granted here for authorized users of LS-LS-INGRID to copy, re-use and modify these examples at will.

5.2 **OCCUPANT** : Occupant (dummy) and seat-belt options

Summary: Up to 5 occupant models may be positioned and oriented. Those with "tree" files may have their component parts oriented interactively.

Seat-belts may be tensioned onto occupants; and retractors, slip-rings, sensors, pre-tensioners and accelerometers may be created.

Required: For dummy positioning at least one dummy model must have been read in with the ***OCCUPANT*** option in the input part of LS-INGRID, see 4.2.2. (For seat-belt positioning, etc alone this is not mandatory.)

Dummy models are normal input decks representing the humanoid dummies used in crash analysis. "Tree" files (see Appendix I) for these models are optional, but if they are defined then dummy component parts (ie limbs) can be oriented in space in an object-oriented fashion. In this way dummies can be positioned correctly in models of vehicles.

Seat-belts need to be positioned accurately over dummy models, and then tensioned into place. A form-finding algorithm is included that will project a crude line onto an existing contact surface, and so fit a seat-belt.

The other crashworthiness items associated with seat-belts: slip-rings, retractors, etc, may also be created and manipulated.

5.2.1 ***SELECT*** : Selecting the "current" dummy

Only one dummy may be "current" for occupant operations so, if you have more than one dummy in the model, you must ***SELECT*** the one to be used. You can screen-pick any element on it, or select it by name. If you only have one dummy it is automatically selected.

This dummy is used for all subsequent **OCCUPANT** operations until a new one is selected, or you leave the **OCCUPANT** menu.

5.2.2 ***LIST*** : List current dummy's attributes

This command will draw a "stick" diagram of the current dummy showing its connectivity, list all its components as defined in its "tree" file, list all contact surfaces defined on it and summarize its current mass.

This command only works if a "tree" file exists for the dummy.

5.2.3 ***SCALE*** : Changing occupant units and scale

Occupant models tend to be standard library files and, as such may be in the wrong units system for your structural model. LS-INGRID is capable of changing model <mass>, <length> and <time> units using this menu. In addition some dummies may be changed from 50th to other percentiles.

Changing model units

Three predefined units systems are built into LS-INGRID:

System name	Mass units	Length units	Time units
SI (Systeme Internationale)	kg	m	s
MM (Modified metric)	Te	mm	s
IN (Imperial)	lb	in	s

Dummies may be changed between these systems using the ***SCALE*** options:

MM_TO_SI	Modified metric to SI;
MM_TO_IN	Modified metric to Imperial;
SI_TO_MM	SI to Modified metric;
SI_TO_IN	SI to Imperial;
IN_TO_SI	Imperial to SI;
IN_TO_MM	Imperial to modified metric;

In addition there is a further ***USER_DEF*** option that allows you to define explicit <mass>, <length> and <time> conversion factors.

In all cases conversions are carried out consistently on all aspects of the model: coordinates, materials, loads, etc.

Changing dummy percentile

For the OASYS Rigid Hybrid III dummy only (type **RH3**) it is also possible to change the size of the dummy from the standard 50th %ile male to other sizes:

TO_5%	Converts to 5th%ile female;
TO_95%	Converts to 95th%ile male;

TO_USER Converts to a user-defined percentile value.

Important notes on percentile conversions:

- Conversions to 5th and 95th%ile dummies are based on the factors in "Size, Weight and Biomechanical Impact Response Requirements for Adult SIze Small Female and Large Male Dummies". Mertz, Irwin, Melvin, Stanaker and Beebe. Report 890756 to the Simulation Standards committee of the American Society of Automotive Engineers.

- Mass, size and inertia of each dummy part is scaled; joint stiffnesses are **not scaled** since no information is available on suitable factors.

- Conversion to "user-defined" percentiles is calculated by fitting a polynomial through the 5th, 50th and 95th factors for the dummies. So, among other things, this implies a gradual sex change between 5th and 95th percentiles! Therefore an intermediate dummy percentile created in this way should be assumed to have approximate properties only.

5.2.4 ***ORIENT*** : Orientation and positioning of dummies

Dummies may be positioned in two ways:

- They may be bodily translated and rotated into position. The whole occupant is so transformed and there is no relative movement of its component parts.

- Individual parts (i.e. limbs) of the occupant may be rotated about their connection parts. This implies a knowledge of the connectivity and hierarchy of the model, and is only possible if a "tree" file has been defined.

Transformation of the whole dummy

TRANSLATE Allows you to move the whole dummy by some vector [dx,dy,dz] which may be defined as follows:

XYZ_USER	Define an explicit [dx,dy,dz] vector;
NEW_H_PT	Define a new [x,y,z] coordinate for the dummy "H-point";
NODE->NODE	Move by the [dx,dy,dz] vector between 2 nodes;
NXY->NXY	Move by the [dx,dy] vector between 2 nodes;
NYZ->NYZ	Move by the [dy,dz] ditto;
NXZ->NXZ	Move by the [dx,dz] ditto;
NX->NX	Move by the [dx] ditto;
NY->NY	Move by the [dy] ditto;

`NZ->NZ` Move by the [dz] ditto;

ROTATE Allows you to rotate the dummy by angles [tx,ty,tz] about one of the following locations:

ORIGIN	Rotate about [0,0,0];
H-POINT	Rotate about current dummy "H-point";
XYZ_USER	Rotate about a defined [x,y,z] coordinate.

Note that you should only rotate about one axis at a time, since the order in which rotations are carried out is significant and compound rotations may not give you what you expect.

REFLECT Allows you to reflect the dummy about a global plane:

XY_PLANE	The global XY plane
YZ_PLANE	The global YZ plane
XZ_PLANE	The global XZ plane

Note that this does not copy the dummy, the original becomes the reflected.

POSITION : Orienting dummy sub-parts (limbs)

It is only possible to do this if a "tree" file has been defined. If it has you can rotate all defined sub-parts about the dummy **local** [X,Y,Z] axis system. Each sub-part will rotate about its connection to the next highest part in the hierarchy, and it will move any subordinate parts (lower in the hierarchy) with it. The "root" part will rotate about the H-Point.

The dummy local axis system is defined in the "tree" file, and convention is that local X is to the dummy right, local Y to its front and local Z vertically up (all directions measured from the H-Point).

Sub-part rotation rules are:

- Rotations only take place about one axis at a time. Otherwise matters would get very confused.

- Rotations are always about the **dummy** (not sub-part) local axis system as defined in the tree file.

- Zero degrees for all sub-parts would cause the dummy to stand "to attention" with his feet pointing downwards.

- The sign convention is right hand screw rule: positive is clockwise when looking down a vector from the origin.

The commands to position dummy sub-parts are:

AXIS To define the local axis about which to rotate. (Default X)
DRAG To "drag and drop" sub-parts using the mouse - see below.
INIT To return all sub-parts to their initial positions.
<part name> Orient the specified sub-part to the specified angle.

All sub-parts defined in the tree file are listed in this menu together with their current angle about the selected rotation axis.

"Drag and drop": moving sub-parts with the mouse

Often the exact angles for sub-parts are not known, and the best method of positioning them is trial and error by "dragging" them across the screen, and then "dropping" them into position. This may be achieved by the ***DRAG*** command in the menu above.

Use the mouse to select a node on the sub-part you want to move. This will be highlighted and, keeping the mouse button depressed, move it to where you want it to be. Once it is in position release the mouse and it will be redrawn in its new position.

The "tree" structure will become immediately apparent when you do this: you will see that sub-parts subordinate to the one you are moving are pulled with it.

Notes:

- Sometimes the sub-part picked is not the one you want. So long as you haven't moved it just release the mouse and try again. If it has moved you can reset it to its orientation before entering the ***DRAG*** menu by using the ***RESET*** command. (Note that this is different to ***INIT*** in the menu above which resets the whole model to its absolute initial orientation.)

- You can change the current axis of rotation by selecting one of ***X_AXIS***, ***Y_AXIS*** or ***Z_AXIS***. This is analogous to using the ***AXIS*** command in the menu above.

- "Drag and drop" is easy once you have got the hang of it, but requires a little practice. It is probably best to experiment with one of the standard dummy models (RH3 is the best) on its own before trying to assemble a complete dummy and vehicle for real.

- It can be slow "dragging" many parts of a complex dummy model around the screen: it puts a load of load on the graphics hardware. For this reason it is best not to try dragging the root part of a dummy, and instead to orient this explicitly in the menu above.

5.2.5 `CONTACT` : Processing contact surfaces on dummy models

If you are going to position seat-belts on a dummy then the belt positioning algorithm needs to know which contact surfaces are significant. You can skip this section if you are not interested in belt positioning.

The following concepts are important:

Contact surface groups

Often it will take more than one contact surface to achieve a structural function. For example a seat belt will run from the shoulder, across the chest and pelvis, and over the upper legs, covering several unconnected materials on its way.

This requires several discrete contact surfaces to be defined: seat-belts require conventional (one sided, non-automatic) contacts which cannot "track" over gaps in the mesh. It is easier to handle this if contacts are grouped by function.

Therefore the **OCCUPANT** section of LS-INGRID contains the concept of a "group" of primitive contact surfaces which may be given a name.
For example "BELT_TO_DUMMY" in the RH3 rigid Hybrid III dummy.

If you are going to position seat-belts you will need at least one such contact surface group, even if it only contains a single contact surface.

Pre-defined contact groups in "tree" files

Usually a standard dummy model will have one or more contact surfaces already defined on it, and the groups of these will be included in its "tree" file. Such groups will appear automatically in this menu and you need take no further action to define them (although you may wish to deselect redundant ones: see below).

Contact surface types

Contact surfaces for seat-belts must be "discrete nodes impacting surface", i.e. type 5; and they must not be "automatic" since it matters whether the belt is outside or inside both to the positioning algorithm and during the analysis. Other pre-defined types, for example for seat-belt or airbag contact, don't matter so much.

This raises the problem that only one side of pre-defined contacts, that of the segments on the dummy, will exist and it is illegal in DYNA3D to have a type 5 contact with no nodes on its "slave" side.

The solution adopted is to pre-define such contacts as "single surface", type 4, and to modify their type to the correct one later.

Processing contact groups

If no contact surface groups have been defined, or those provided are inadequate, you will have to create some. You can also deselect unwanted (pre-defined) groups so that they don't appear in the output model. The commands to do this are:

SELECT To select and de-select existing contact surface groups for dummy operations and subsequent output.

GROUP To make groups out of existing contact surfaces. Generally these will be pre-defined, but you can create them with the main menu ***CONTACT*** options.

SKETCH Lets you sketch contact groups and individual surfaces in them on the current image.

SUMMARY Will summarize all contact group contents and their selection status to the dialogue, so it will not destroy the current image.

LIST Lists more detailed information about all contact surface groups and their contents to the graphics area, so destroying the current image.

EXPLAIN Provides on-line help on using this menu.

The process is straightforward:

- If you need to define or modify a contact surface group you do so with the ***GROUP*** command. This lets you create and delete groups, and then add or remove individual contact surfaces from them.

- Then you ***SELECT*** those you wish to be used for this dummy. By default all groups are selected: you can leave this as it is, although it will be inefficient during the analysis phase to have redundant contacts active.

- At any time you can use the ***SKETCH*** option to visualize what you have done.

- When you have finished you can return to the main **OCCUPANT** menu and carry on with belt fitting.

5.2.6 *BELTS* : Seat-belt positioning and processing

This sub-menu handles the positioning and creation of seat-belts, and also the creation of all the related occupant restraint items.

It is assumed that the reader understands occupant restraints systems, and has access to a current LS-DYNA3D user manual explaining what input is required for each of these items.

The options available are:

SEAT_BELTS	Create and position seat-belt elements. Uses a form-finding algorithm to tension belt on dummy contact surface(s).
MATL_BELTS	Define, modify and list seat-belt materials and their properties.
SLIP_RINGS	Define slip-rings ("D" rings) for seat-belts.
RETRACTORS	Define retractors (inertia reels) for seat-belts.
SENSORS	Define sensors for retractors and pre-tensioners.
PRE_TENS	Define pre-tensioners for seat-belts.
ACCELER	Define accelerometers at nodes on rigid bodies.
SKETCH	Sketch all items of the types above on the current image.
STATUS	List the status of all items above to the dialogue area.

To create and position a seat-belt

Preliminary actions to go through:

- Create a belt material, if one does not exist already, using ***MATL_BELT***. You will need at least one current load-curve to do this so, if you haven't got any, use the main menu **LCURVES** option to make some first.

- Position your dummy so that you have a good view of the parts on which you want to fit a belt. It is probably best to blank everything except the dummy and those parts of the structure that you need.

- Make sure that you know in advance exactly where the belt end-points are on the vehicle structure, and ensure that the nodes at these points are visible since you will need to screen-pick them. Alternatively make sure that you know their node labels.

(a) Create a line giving an approximate seat-belt path

Use ***SEAT_BELTS*** to enter the seat-belt-processing menu, and select option ***CREATE***. This will present the menu:

NODE->NODE	Pick a line from node to node mode;
USER_XYZ	Define global [x,y,z] points for line mode;
REJECT_LAST	Reject last point selected;
RESTART	Throw away current line and start again;
DONE	Accept current line and continue.

What you are going to do is to create a crude line that lies roughly where the seat belt needs to go. It is made up of any number of straight line segments between points, and you define it by giving the points coordinates. The line and its points have no structural significance, they are simply used as a "construction line" to define the initial belt shape.

The points on the line can be defined in two different ways:

NODE->NODE Lets you pick a series of nodes with the mouse. Points are created at the nodal coordinates.

USER_XYZ Lets you type in a series of [x,y,z] coordinates for the points. This is the default mode.

You may toggle between the two modes at will, using ***REJECT_LAST*** to cancel any points you have got in the wrong place. When the line is complete use ***DONE*** to move on to the next stage.

When creating a line you should note the following:

- It is not necessary to trace a very accurate path over the dummy. Line segments of (say) 0.5m are quite acceptable so long as you place points on places where significant angular change will occur.

- Line segments do not represent characteristic belt element length at this stage: they are sub-divided later.

- The line may pass through parts of the dummy at this stage, when tensioned later it will be projected out onto the appropriate contact surface. However try not to create massive penetrations since the contact projection algorithms may not be able to handle these.

- It is acceptable to define a complete belt in one pass, for example when defining a 3-point lap and diagonal belt you can define the complete belt: say from retractor, through slip-ring, over torso, to buckle point, over pelvis and to floor mounting, in one line.

(b) Define a characteristic length for belt elements

Once the line is complete, signified by using ***DONE*** above, you will drop into the menu that defines the characteristic length of the seat-belt elements to be used. Typical lengths might be 50mm, and you will be given a choice of suitable lengths, but you may choose any length.

Once you choose a length each line segment will be linearly sub-divided into the closest possible multiples of the length you chose. These sub-divisions will be sketched onto the crude line, and you will be asked whether they are satisfactory. You can respond:

OK	Sub-divisions are acceptable, continue to next step.
SK_LINE	Redraw the line (useful if you change views);
CH_CHAR	Change characteristic length and repeat;
CH_BASIC	Change basic line: modify points;
RESTART	Start from scratch again with a new line.

Notes:

- Characteristic belt length may have an influence on model minimum time-step size: very short belt elements can be the critical ones. So don't choose ridiculously short values or you may regret it!

(c) Define which contact surface(s) to use for belt fitting

Generally you will select one or more contact groups here, but it is possible to limit this to individual contact surfaces within a group.
(See 5.2.5 for a description of contact surface groups.)
The belt will use these surfaces when form-finding its path.

The commands in this sub-menu are:

S_ALL	Select all contact surface groups;
D_ALL	De-select all contact surface groups;
INDI	Enter sub-sub-menu to select individual surfaces;
DONE	Finish surface selection process and continue;
STATUS	List current selection status in dialogue area;
OPTIONS	Further listing and sketching options;
<group name #1>	
<group name #2>	
etc	

Notes:

- Form-finding will slow down as more contact surfaces are made active. So choose only those you actually need.

(d) Tension the belt ("form-find") onto the dummy

Form-finding positions, or tensions, the belt onto the dummy by adjusting the crude line you have defined into a more realistic path.

The rules the form-finding algorithm uses are:

- Try to reduce the line to a straight line between fixed points.
- To do this adjust intermediate points to lie on the straight line between their neighbours, but:
- Always project any point that penetrates a contact surface back onto that surface.
- Carry out this procedure iteratively until the change between successive iterations drops below a preset minimum value.

Preliminary action: fixing intermediate points

Before you proceed with tensioning you must define all points on the crude line that are fixed: such points do not move and are typically used at slip-rings.

By default the first and last points are always fixed, but any intermediate point may also be fixed using the options:

FIX	To fix points
UNFIX	To unfix (free) points

Whenever the line is sketched "fixed" point symbols have a solid green triangle drawn inside them. And within the ***FIX*** and ***UNFIX*** menus you can list fixed points and re-sketch the line at any time.

Before form-finding proceeds a check is made for angular change between all line segments. If any angle is found to be less than 90 degrees you will be asked whether you want to ***FIX*** it since, typically, such geometry only occurs at slip-rings.

Commencing with form-finding

The ***GO*** option causes form-finding to commence. At each iteration the new belt line is redrawn so you can see how the line is moving. The iteration number and current convergence tolerance are also listed in the dialogue area.

Progress of the form-finding operation

The iteration process will continue until the current convergence tolerance is reached. This is usually achieved in under 50 cycles.

However it also halts if the cycle number is a multiple of 25 to allow you to abort it if it is clearly not converging. To continue with convergence type ***GO*** again, otherwise change something using the guidance in (f) below.

(e) Finally convert the converged line into seat-belt elements

When you are happy with the path of the line type ***DONE*** to pass to the final stage which is to convert it into actual belt elements.

The options here are:

SIMPLE	Use DYNA3D "seat-belt" elements only;
MIXED	Mix "seat-belt" elements with shells. This option has not yet been implemented in release 3.4A.

Since only the "simple" operation is currently available all you have to do is to define a belt material for the elements from the list given, and the remainder of the procedure is automatic:

- The requisite number of seat-belt elements are created;

- All of the nodes on the belt are made into "slave" nodes to the type 5 (discrete nodes impacting surface) contact surfaces in the contact group(s) selected for this belt.

- If the contact surface in the group(s) are not type 5 they are converted to be so: you are warned if this happens.

- The newly created seat-belt elements are sketched on the mesh.

At this stage you have successfully created and positioned a seat-belt on your dummy, and you will return to the **SEAT BELTS** menu.

(f) Common form-finding problems and their solutions

Sometimes things go wrong during the form-finding operation. This section lists known examples of this and their solutions:

The shape fails to converge

Sometimes the shape will "hunt" repeatedly between a small family of shapes in a particular area, and never reach the convergence tolerance.

- Often this doesn't matter: there is no "right" shape for a belt, so if the result is close enough then accept it as it is.

- If it does matter you can adjust one or more points manually using the ***MOVE*** option. Shift them to a new position, then restart the convergence procedure by typing ***GO*** again.

- If this still doesn't work you may have to go back and re-define the line in a slightly new position using the ***CH_BASIC*** command. This takes you back to step (a) above.

The shape converges too soon

Sometimes it converges too early, and you feel that it could move a bit further into a better position.

- Try tightening the convergence tolerance from the default of 1.0e-5 to something smaller using the ***CH_TOL*** command.

- Alternatively move some points to where you want them to end up using ***MOVE*** as described above.

The line "punches through" a contact surface

Where the surface it crosses is concave it is possible for the belt to fall through the gaps between contact segments. This is a geometric problem familiar to any user of contact surfaces in DYNA3D, and it manifests itself here since a similar contact algorithm is used.

- Try increasing the maximum penetration depth with ***MAX_PEN***. This may catch nodes that fall through gaps.

- Try to move the offending nodes away from the concave area of mesh using ***MOVE***, or perhaps by redefining the line.

However this problem may persist and, if it does and is it clearly due to concavity, you have no choice but to remesh the offending area to reduce the angular change between adjacent contact segments: LS-INGRID is simply anticipating what is likely to happen when you run this problem in DYNA3D.

Modifying existing seat-belt elements

Once some seat-belt elements have been created you can use the other options in the **SEAT BELT** menu:

MODIFY	Lets you add and modify belt element characteristics;
DELETE	Permits seat-belt elements to be deleted.
SK<option>	Sketches belt entities and their attributes.

Of these the ***MODIFY*** option requires more explanation. Its sub-menu is:

JOIN	Joins separate belt sections together;
NODE_POS	Moves nodal coordinates;
MATERIAL	Lets you change belt material numbers;
SLACK	Lets you set initial slack length of belt elements;
CONTACT	Lets you add/remove belt nodes to/from contact surfaces.
SK<option>	Sketching options for belt-related data

In more detail:

- To ***JOIN*** two unconnected sections of belt select an element from each section, then choose which nodes on each element are to be merged. The topology of the belt elements will be updated so that the two elements share the common node. For this to work to two nodes should be nearly coincident.

- ***NODE_POS*** to move nodal coordinates may be useful prior to a ***JOIN***, or when adjusting geometry at retractors and slip-rings. You simply define new coordinates.

- ***MATERIAL*** modifies the material number used for belt elements, it does not modify the material properties themselves: you can do that using ***MATL_BELTS*** in the main **OCCUPANT, BELTS** section.

- ***SLACK*** sets the initial slack length of belt elements. This is normally zero.

- ***CONTACT*** allows you to adjust manually which belt nodes are slaves to which contact surfaces. The belt creation routines make all nodes on a belt slaves to all surfaces used for form finding, but it is efficient only to have nodes that are likely to cross a surface actually in its node list.

- The various ***SKETCH*** options, common throughout the seat-belt processing section, may be used to sketch belt element attributes on the current mesh.

MATL_BELTS : Processing belt materials

The use of this option to create belt materials has been mentioned previously, since it is a necessary precursor to creating belt materials. The seat-belt material processing options are:

CREATE	To define a new material from scratch;
COPY	To define a new material by copying an existing one;
MODIFY	To modify an existing material
DELETE	To delete an existing material
SK<option>	To sketch belt and material attributes
LIST	To list a material's properties in the dialogue area
STATUS	To summarize all belt material properties in the dialogue area.

The processing of seat-belt materials is very simple since they need:

- Two load-curves: for loading and unloading;
- A mass per unit length;
- A minimum length.

Therefore you must have at least one load-curve current in order to create a belt material. If you haven't got any use **LCURVES, CREATE** from the main menu to create some.

In addition you cannot ***DELETE*** a belt material that is referenced by any belt elements. You must delete the belt elements first, or make them reference a different belt material.

Processing Slip_rings,Retractors, Sensors, Pre-tensioners and Accelerometers

All these items have the same sub-menu:

CREATE	To create the item;
MODIFY	To modify its attributes;
DELETE	To delete the item;
SKETCH	To sketch belt items;
STATUS	To list item status.

Usage is simple since these all lead you through a series of menus that allow you pick the nodes, belt elements and other values required in each case.

Slip-rings

- Require that at least two belt elements exist;
- Require a node at the common location of the belt elements that is **not** on the belt itself;
- Have a friction coefficient;

Retractors

- Require at least one belt element to exist;
- Require a node at the retractor location that is **not** on any belt element;
- Require at least one sensor to exist, and can use up to four;
- Require at least two load curves for loading and unloading;
- Have a trigger time delay, a pull-out distance, and a fed-length;
- May have extra seat-belt elements specified inside them.

Sensors

- May be of 4 different types, type 2 requiring at least one retractor to exist;
- Must have their dynamic relaxation status defined;
- Have further type-dependent attributes.

Pre-tensioners

- May be of three different types;
- Require at least one retractor and one load curve to exist for type 2;
- Require at least one spring to exist for type 3;
- Have further type-dependent attributes.

Accelerometers

- Are defined by three nodes which **must** be on a rigid body.

General notes on these types:

- Some items depend upon the existence of others, as described above. This is checked, and you will not be permitted to create an item unless those it depends on also exist.

- Likewise you will not be allowed to delete an item that is depended upon by something else.

- In the main menu **BLANK** option seat-belts and their related types are lumped together under the ***BELT_etc*** heading, and you need to drop into this sub-menu to select them. This is to prevent the menu of element types getting excessively cluttered.

5.2.7 Worked Examples

Appendix III contains a worked example that demonstrates the positioning of a dummy in a vehicle, and the subsequent fitting of a seat-belt.

5.3 ***ORIENT*** : Translation, rotation and scaling of entities

Summary: The whole model, (or any sub-set of it defined by parts, materials, contact surfaces or individual elements), may be rotated, translated, reflected or scaled.

Transformations take place on all transformable attributes.

Required: <none>

5.3.1 The transformation options in the main **ORIENT** menu

TRANSLATE Allows you to specify a translation by some vector [dx,dy,dz] which may be defined as follows:

XYZ_USER	Define an explicit [dx,dy,dz] vector;
NODE->NODE	Move by the [dx,dy,dz] vector between 2 nodes;
NXY->NXY	Move by the [dx,dy] vector between 2 nodes;
NYZ->NYZ	Move by the [dy,dz] ditto;
NXZ->NXZ	Move by the [dx,dz] ditto;
NX->NX	Move by the [dx] ditto;
NY->NY	Move by the [dy] ditto;
NZ->NZ	Move by the [dz] ditto;

ROTATE Allows you to specify a rotation by angles [tx,ty,tz] about one of the following locations:

ORIGIN	Rotate about [0,0,0];
XYZ_USER	Rotate about a defined [x,y,z] coordinate;
NODE_POS	Rotate about a node position.

Note that you should only rotate about one axis at a time, since the order in which rotations are carried out is significant and compound rotations may not give you what you expect.

REFLECT Allows you to define a reflection about a global plane:

XY_PLANE	The global XY plane
YZ_PLANE	The global YZ plane
XZ_PLANE	The global XZ plane

Note that this does not copy the entities, the original becomes the reflected.

Once you have specified a transformation you must then define what it is to act upon. The options are:

ALL	The whole model;
PART	Selected part(s) only;
MAT	Selected material(s) only;
SURFACE	Selected contact surface(s) only;
<el types>	Ranges of selected element types only.

Once you have chosen these the transformation is applied.

Everything that can be transformed is, but there are some items that can not be. These are:

- Nodal and rigid body restraints;

- Applied motion (disp/vel/acc) boundary conditions;

- Some arcane and little-used options.

5.3.2 The ***SCALE*** option for changing units and scale works as follows

Three predefined units systems are built into LS-INGRID:

System name	Mass units	Length units	Time units
SI (System International)	kg	m	s
MM (Modified metric)	Te	mm	s
IN (Imperial)	lb	in	s

Items may be changed between these systems using the ***SCALE*** options:

MM_TO_SI	Modified metric to SI;
MM_TO_IN	Modified metric to Imperial;
SI_TO_MM	SI to Modified metric;
SI_TO_IN	SI to Imperial;
IN_TO_SI	Imperial to SI;
IN_TO_MM	Imperial to modified metric;
USER_DEF	User defined <mass>, <length> and <time> factors.

As with transformations you must define what is to be scaled, but in this case you only choose the whole model or selected parts. Again all entities that can be scaled are.

5.4 ***CONTACT*** : Creation and processing of contact surfaces

Summary: Contact surfaces may be created, modified and viewed. Contact control attributes may be listed and modified, and contacts may be selected/deselected for output.

In release 3.4A contacts may only be created by material, not by individual segments.

Required: <none>

The options in the main **CONTACT** menu are:

CREATE	Create a new contact surface;
MODIFY	Modify an existing contact surface;
SELECT	Select a surface for eventual output;
DESELECT	Deselect a surface to inhibit its output;
SKETCH	Sketch existing surfaces on the current image;
LIST	List information on a surface to the graphics area;
STATUS	List a summary of all surfaces to the graphics area.

5.4.1 ***CREATE*** : Creating a new contact surface

Firstly you are required to define some global contact parameters:

- Type: currently from 1 to 23;
- Static and dynamic friction coefficients;
- Penetration check options;
- Force output to **.CTF** file flags;
- Slave and master side scale factors (slave only for single-surface);

Then you must enter the list of materials and optional contact box limits for slave and master sides.

When this has been done the contact surface is created as the next highest free surface number, and its component parts are sketched on the screen.

Notes:

- Material lists are used to define segments on faceted contact types, and lists of nodes on the slave side of "discrete node" contact types.

- It is possible to enter the same material on both sides of a surface. This is not necessarily an error, although in most cases it will be wrong.

5.4.2 ***MODIFY*** : Modifying an existing contact surface

Firstly you are required to specify which surface you want to modify, and then you can change any of the following options:

TYPE Change contact type. Not all categories of contact type are interchangeable, so use with care.

MATERIALS You can add or remove materials from master or slave sides of the surface.

FRICTION You can modify the static and dynamic coefficients of friction of the surface, and also the friction decay coefficient and viscous criterion.

STIFFNESS Master and slave penalty stiffness can be modified, and the contact formulation can be toggled between penalty force and soft constraint methods.

BIRTH/DEATH The "birth" and "death" times of the contact can be set.

LC_CONTROL For rigid-rigid contacts types 19 to 21 only the load curves and calculation methods can be modified.

AIRBAG_LC For special airbag contact type a13 only the load curve controlled thickness option can be set.

SEARCHING The searching depth, bucket-sort frequency, force update frequency and penetration parameters can be set.

FAILURE For type 8 and 9 contacts only the weld failure force/stress parameters and exponents can be set.

SLAVE_BOX Allows the slave side model box definition to be modified.

MASTER_BOX Allows the master side model box definition to be modified.

The other ***MODIFY*** options allow you to finish and see what you have done:

NEW_SURFACE Terminates modifications on the current surface and lets you select a new one to modify;

LIST Lists the attributes of the current surface in the graphics area;

SKETCH Sketches the current surface on the current display;

DONE Terminates the contact modification process.

5.4.3 ***SELECT/DESELECT*** : (De)selecting contact surfaces for output

In LS-INGRID 3.4A it is not possible to delete a contact surface once it has been defined, however it can be ***DESELECTED*** at any time which will have the effect of suppressing it when the output file is written.

It may be ***SELECTED*** or ***DESELECTED*** back and forth at any time, the effect only becomes permanent when the output phase is entered.

A "deselected" contact surface will not be displayed by normal graphics commands, however it may still be "sketched" within the **CONTACT** section.

5.4.4 Viewing and listing contact surface data

The final commands in the ***CONTACT*** menu allow you to visualize and list your contact surfaces. They are:

SKETCH Will "sketch" any surface(s) on the current model. In the case of surfaces defined by materials the nodes/elements of the material as appropriate, together with any finite model box, will be shown.

LIST Will "list" the attributes of the selected surface in the graphics area of the screen.

STATUS Will give a summary of all contact surfaces in the graphics area of the screen.

Notes:

- Explicitly segmented contacts have arrows drawn on them showing their outward normal direction.

- Discrete node contacts have circles drawn round the relevant nodes.

- The ***SKETCH*** option above is the only way to visualize contacts defined by material, since it draws the relevant elements/nodes. These are not currently drawn by the main graphics commands in LS-INGRID.

5.5 ***LCURVES*** : Creation and processing of load curves

Summary: Load curves may be created, modified, listed and plotted. In release 3.4A load curves can **not** be deleted.

The options in the main **LOAD CURVES** menu are:

CREATE	Create a new load curve;
MODIFY	Modify an existing curve;
PLOT	X-Y plots of existing curves;
SUMMARY	Summarize all curves in teh graphics area;
LIST	List the key attributes and points of a curve.

5.5.1 ***CREATE*** : Creating a new load-curve

This command allows you to create new load-curves. The curve created is always the next highest free number. The sub-menu here is:

KEY_IN	Type in [x,y] data pairs for each point;
FILE	Read in points from an ASCII file;
COPY	Copy an existing curve.

<u>Keying in points:</u>

This is straightforward: simply type in values. Options are:

DONE	To finish entry and store curve;
LIST	To list the curve so far on the screen;
PLOT	To plot the curve so far on the screen;
REJECT	To reject the last point(s);

<u>Reading from a file:</u>

Give the filename you want to read from. You can see what files are available on disk using the options:

LIST_FILES	To list all files matching the current "filespec" below;
FILESPEC	To set a new "filespec" search path and string.

Data is read as <x value> <y value> using Fortran "free" format. Any line which generates a Fortran "read error" is skipped, and reading continues until the end of the file is found. Therefore any lines that do not contain plain numbers, for example title strings and comments, will be skipped.

Copying an existing curve:

The data from an existing curve are copied into the new one.

Curve attribute settings:

As well as data load curves in DYNA3D contain the following attributes:

- A flag stating whether the curve if for transient or dynamic relaxation analyses, or both. (Default transient only.)
- A scale factor on each axis. (Default each 1.0)
- An offset for each axis. (Default 0.0)

When curves are created using ***CREATE*** all these attributes are set to their default values.

5.5.2 *MODIFY* : Modifying an existing load-curve

Any existing load curve can be modified. The sub-menu here is:

POINTS	Modify the [x,y] data points in the curve;
SCALE	Modify the x and y scale factors;
OFFSET	Modify the x and y offset values;
DRFLAG	Modify the transient/dynamic relaxation flag.

Modifying data points

Points may be overwritten or deleted, but in version 3.4A they may not be inserted. Options are:

INSERT	To insert data points (Not available in 3.4A);
DELETE	To delete data points;
OVERWR	To overwrite data points with new values;
DONE	To finish changes;

Changing scale factors

The current X and Y scale factors are displayed and you are invited to enter new values.

Changing offsets

The current X and Y offsets are displayed (if non-zero) and you are invited to enter new values.

Changing the transient/dynamic relaxation flag

The current flag is displayed and you are invited to enter a new value from the options:

TR_ONLY	Curve is active during transient analysis only;
DR_ONLY	Curve is active during dynamic relaxation only;
BOTH	Curve is active during both analysis phases.

Listing curve points and attributes

You can list all data points, and all the settings of these attributes, when you ***LIST*** a curve.

5.5.3 *PLOT* : Plotting existing load-curves

Any existing load curve(s) can be plotted in X-Y graphical form. The sub-menu here is:

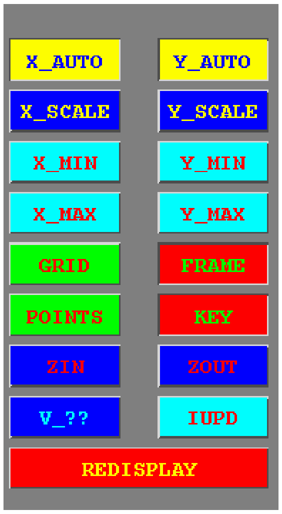
SELECT	Select the curve(s) to plot;
TITLE	Give a new plot title;
X_LABEL	Give a new X (horizontal) axis label;
Y_LABEL	Give a new Y (vertical) axis label;
SUMMARY	Summarize all curves in the graphics area;
LIST	List the key attributes and points of selected curve;
F_COLOR	Frame color: set a new color for the frame around the plot;
G_COLOR	Grid color: set a new color for the plot grid.

The process is very simple:

- ***SELECT*** the curve(s) to be plotted, and they will be drawn immediately.

- Tune the attributes of the plot using the menu options and the special plotting options mapped for this purpose.

The "viewing control" box mapped during curve plotting



The commands in this box control curve plotting attributes, and have the following functions:

X_AUTO, Y_AUTO Set automatic X and Y scaling respectively;
X_SCALE, Y_SCALE Set explicit factors on the X and Y scales;
X/Y_MIN, X/Y_MAX Set explicit upper and lower X and Y axis limits;

GRID Controls whether or not a grid is superimposed;
POINTS Controls whether or not symbols are drawn at points;
FRAME Controls whether or not a border is drawn around the plot;
KEY Controls whether or not a key of curve ids is drawn;

ZIN, ZOUT Uses the cursor to zoom in and out on the plot

IUPD Sets the "instant update" switch on or off;
V_?? Gives on-line help on all these options;
REDISPLAY Redraws the current plot.

By far the easiest way to learn to use these is to turn the ***IUPD*** switch on, so that changes have an immediate effect, and then to experiment.

5.5.4 Listing and summarizing load-curve data

The ***SUMMARY*** and ***LIST*** options may be used to send summary data for all curves and detailed information on selected curves, to the screen.

5.6 *BOUNDS* : Processing of Boundary Conditions

Summary: Nodal and rigid body initial velocities, restraints, applied velocity/displacement/acceleration boundary conditions may be applied. All such boundary conditions may be listed and sketched.

Base accelerations and angular velocities may be applied.

The *BOUNDS* menu is:

ND_REST	Processing of nodal restraints (SPCs)
ND_INIV	Processing of nodal initial velocities
ND_DVA	Processing of nodal applied disp/vel/acc boundary conditions
RB_REST	Processing of rigid body restraints
RB_INIV	Processing of rigid body initial velocities
RB_DVA	Processing of rigid body applied disp/vel/acc boundary conditions
BASE_AC	Processing of base accels/angular velocities on whole model.

All items in this menu share some common characteristics:

Degree of Freedom (DoF) codes

They may be applied in [X,Y,Z] translational and rotational degrees of freedom. In LS-INGRID these DoFs are referred to by numeric codes and direction codes as well as names. The numbers and direction codes used are:

X Translation	:	1	dX
Y Translation	:	2	dY
Z Translation	:	3	dZ
X Rotation	:	4	tX
Y Rotation	:	5	tY
Z Rotation	:	6	tZ

These DoF codes are combined in some contexts to make a composite code, for example 135 refers to [dX,dZ,tY], i.e. translation in X and Z, and rotation about Y.

What Rotational Degrees of Freedom Mean

Users should note that a rotational degree of freedom for an item in DYNA3D implies a spin of that item itself about some local axis acting through its centroid. Note particularly that in the case of a node it does **NOT** imply angular velocity of the node about some remote axis.

Global versus Local axis systems

By default all degrees of freedom are assumed to be in the global cartesian coordinate system. However it is possible in some contexts for these DoFs to be defined in some other, "local" coordinate system.

5.6.1 ***ND_REST*** : Applying nodal restraints

Nodal restraints, (sometimes referred to as "Single Point Constraints", or SPCs), may be applied individually to the any or all of the six degrees of freedom of nodes. The menu options are:

ADD	To add restraints
REMOVE	To remove restraints
SKETCH	To sketch all current restraints on the display
LIST	To list restraints at the selected nodes

The ***ADD*** and ***REMOVE*** options both require you to define which DoFs, in what coordinate system, you want to add/remove. The sub-menu to select these is:

```
1       (X translation)
2       (Y    "    "  )
3       (Z    "    "  )
4       (X rotation)
5       (Y  "   "  )
6       (Z  "   "  )
GLOBAL  (Global system)
LOCAL   (Local system)
DONE    (<current code>)
```

Use this by:

- Selecting those DoFs you wish to add/remove. As each one is selected the current code (e.g. 135) will be appended to the current ***DONE*** string.

- Selecting either ***GLOBAL*** or ***LOCAL*** coordinate systems. In the "local" case you must specify which existing coordinate to use.

- Finally using ***DONE*** to finish DoF selection and proceeding actually to add/remove the restraints at the selected nodes.

Note that:

- ***ADD***ing restraints to a node in a DoF at which a restraint already exists will not matter - the definition is not duplicated.

- To delete a restraint you must ***REMOVE*** it explicitly.

5.6.2 ***ND_INIV*** : Applying nodal initial velocities

Nodal initial velocities may be applied individually to the any or all of the six degrees of freedom of nodes. They may only be defined in the global cartesian coordinate system. The menu options are:

VALUES	To define the current [Vx,Vy,Vz] and [Rx,Ry,Rz] vectors;
MASK	To define a "mask" defining which DoFs are to be updated;
APPLY	To apply the current values to selected nodes;
SKETCH	To sketch all current restraints on the display;
LIST	To list the velocity vectors at the selected nodes;
CURRENT	To list the current "values" vectors.

This process is very similar to applying nodal restraints:

- You define the individual values of velocity in each degree of freedom. Each value may be set separately, the default in all cases being zero.

- You apply them to the selected nodes, subject to the current "mask" as described below.

MASK : Applying only a subset of all 6 degree of freedom

The ***MASK*** command exists because, unlike restraint codes, a (default) velocity of zero in a given DoF will cause that component of nodal velocity to be set to zero: it does **NOT** mean "leave the current value unchanged".

This will be a nuisance when you want to update (say) the Vy and Vz components of all nodal velocities without affecting any other components. You can use the ***MASK*** command to make this possible by "masking off" only those degrees of freedom you want to change.

By default the "mask" is set to <123456>, meaning that all DoFs will be updated. In the example above, you would set it to <23>, and then only the Vy and Vz components of velocity would be affected by the current velocity vectors.

Interaction with rigid-body initial velocities

Theoretically nodal initial velocities should be ignored for nodes on a rigid body. In practice DYNA3D integrates such nodal mass and velocity vectors and generates an equivalent rigid-body motion applied at the rigid body centroid.

The consequences in DYNA3D of defining both nodal and rigid-body initial velocities for nodes on rigid bodies are undefined: take care!

5.6.3 ***ND_DVA*** : Applying prescribed displacement/velocity/acceleration at nodes

Prescribed displacement, velocity or acceleration boundary conditions at nodes may be defined in a variety of ways. The user must give:

- The load-curve defining variation with respect to time;
- The type of boundary condition (Disp, Vel or Accel);
- The degree(s) of freedom to be applied.

For some degrees of freedom that represent motion about a vector the user must also supply the vector.

The menu options are:

APPLY	To apply the currently defined values to nodes;
DELETE	To remove values from nodes;
VALUES	To define the values to be applied;
SKETCH	To sketch all current restraints on the display;
LIST	To list boundary conditions at the selected nodes;
CURRENT	To list the current "values" attributes.

Use this option as follows:

Define the conditions to be applied with ***VALUES***

The sub-menu here is:

LCURV	(Required)	Define the load-curve to be used.
MODE	(Required)	Define displacement, velocity or acceleration.
DOF	(Required)	Define which Degree(s) of freedom to load.
FACTOR	(Optional)	Define factor on load-curve values (Def = 1.0).
VECTOR	(Optional)	Define vector for some DoF directions.

You will be warned if you have not set enough items for application to go ahead.

Apply/remove the "values" using ***APPLY*** or ***DELETE***

Once you have defined enough "values" you can apply them to the selected nodes with ***APPLY***, or remove them using ***DELETE***. For each node there can only be one set of applied boundary conditions in a given degree of freedom, so the rules followed are:

- ***APPLY*** will overwrite the old definitions with the new ones;
- ***DELETE*** will delete boundary conditions matching the nodal DoF.

5.6.4 ***RB_REST*** : Applying rigid body restraints

Rigid body restraints are applied at the centroid of the given rigid body, and each degree of freedom is separate. Rigid body restraints impose restraints on all nodes on the rigid body, and they will "win" if separate nodal restraints have been applied. The menu options are:

ADD	To add restraints
REMOVE	To remove restraints
SKETCH	To sketch all current restraints on the display
LIST	To list restraints on all restrained rigid bodies

As with nodal restraints the ***ADD*** and ***REMOVE*** options both require you to define which DoFs, in what coordinate system, you want to add/remove. The sub-menu to select these is identical to that used for nodes:

```
1       (X translation)
  etc to
6       (Z   "    "  )
GLOBAL  (Global system)
LOCAL   (Local system)
DONE    (<current code>)
```

Use this by:

- Selecting those DoFs you wish to add/remove. As each one is selected the current code (e.g. 135) will be appended to the current ***DONE*** string.

- Selecting either ***GLOBAL*** or ***LOCAL*** coordinate systems. In the "local" case you must specify which existing coordinate to use.

- Finally using ***DONE*** to finish DoF selection and proceeding actually to add/remove the restraints on the selected rigid bodies.

Note that:

- ***ADD***ing restraints to a rigid body in a DoF at which a restraint already exists will not matter - the definition is not duplicated.

- To delete a restraint you must ***REMOVE*** it explicitly.

- You should be aware that restraining a rigid body will implicitly restrain the nodes on that rigid body in one or more degrees of freedom.

- Rigid body restraints act about the current centroid. For rotational degrees of freedom this is significant.

5.6.5 ***RB_INIV*** : Applying rigid body initial velocities

Rigid body initial velocities may be applied individually to the any or all of the six degrees of freedom of rigid bodies. They may only be defined in the global cartesian coordinate system. The menu options are:

VALUES	To define the current [Vx,Vy,Vz] and [Rx,Ry,Rz] vectors;
MASK	To define a "mask" defining which DoFs are to be updated;
APPLY	To apply the current values to selected rigid bodies;
SKETCH	To sketch all current restraints on the display;
LIST	To list the non-zero velocity vectors on all rigid bodies;
CURRENT	To list the current "values" vectors.

This process is identical to applying nodal initial velocities:

- You define the individual values of velocity in each degree of freedom. Each value may be set separately, the default in all cases being zero.

- You apply them to the selected rigid bodies, subject to the current "mask" as described below.

MASK: Applying only a subset of all 6 degree of freedom

The ***MASK*** command exists because, unlike restraint codes, a (default) velocity of zero in a given DoF will cause that component of velocity to be set to zero: it does **NOT** mean "leave the current value unchanged".

This will be a nuisance when you want to update (say) the Vy and Vz components of selected rigid body velocities without affecting any other components. You can use the ***MASK*** command to make this possible by "masking off" only those degrees of freedom you want to change.

By default the "mask" is set to <123456>, meaning that all DoFs will be updated. In the example above, you would set it to <23>, and then only the Vy and Vz components of velocity would be affected by the current velocity vectors.

Interaction with nodal initial velocities

Theoretically nodal initial velocities should be ignored for nodes on a rigid body. In practice DYNA3D integrates such nodal mass and velocity vectors and generates an equivalent rigid-body motion applied at the rigid body centroid.

The consequences in DYNA3D of defining both nodal and rigid-body initial velocities for nodes on rigid bodies are undefined: take care!

5.6.6 ***RB_DVA*** : Applying prescribed disp/vel/accel for rigid bodies

Prescribed displacement, velocity or acceleration boundary conditions for rigid bodies may be defined in a variety of ways. The user must give:

- The load-curve defining variation with respect to time;
- The type of boundary condition (Disp, Vel or Accel);
- The degree(s) of freedom to be applied.

For some degrees of freedom that represent motion about a vector the user must also supply the vector. In all cases action is applied at the rigid body centroid.

The menu options are:

APPLY	To apply the currently defined values to rigid bodies;
DELETE	To remove values from rigid bodies;
VALUES	To define the values to be applied;
SKETCH	To sketch all current restraints on the display;
LIST	To list boundary conditions on all affected rigid bodies;
CURRENT	To list the current "values" attributes.

Use this option as follows:

<u>Define the conditions to be applied with ***VALUES***</u>

The sub-menu here is:

LCURV	(Required)	Define the load-curve to be used.
MODE	(Required)	Define displacement, velocity or acceleration.
DOF	(Required)	Define which Degree(s) of freedom to load.
FACTOR	(Optional)	Define factor on load-curve values (Def = 1.0).
VECTOR	(Optional)	Define vector for some DoF directions.

You will be warned if you have not set enough items for application to go ahead.

<u>Apply/remove the "values" using ***APPLY*** or ***DELETE***</u>

Once you have defined enough "values" you can apply them to the selected rigid bodies with ***APPLY***, or remove them using ***DELETE***. For each rigid body there can only be one set of applied boundary conditions in a given degree of freedom, so the rules followed are:

- ***APPLY*** will overwrite the old definitions with the new ones;

- ***DELETE*** will delete boundary conditions matching the current DoF.

5.6.7 ***BASE_AC*** : Applying "base" accelerations and angular velocities

"Base" accelerations act on the whole model, and are applied as a force at nodes proportional to their nodal mass. They may be applied as [aX,aY,aZ] components in the global cartesian coordinate system, with values varying with time using a load-curve. They would normally be used to apply "field" loads such as gravity.

"Base" angular velocities define the centripetal forces on nodes due to rotation in one of the global cartesian [x,y,z] axes about a point. The angular velocities are assumed to be in radians per second, and the configuration assumed is the current deformed geometry. They are normally used to simulate the loads applied in spinning systems such as turbines.

The sub-menu here is:

X_ACC	Base acceleration in X direction
Y_ACC	Base acceleration in Y direction
Z_ACC	Base acceleration in Z direction
X_VEL	Angular velocity about X axis
Y_VEL	Angular velocity about Y axis
Z_VEL	Angular velocity about Z axis
DELETE	Delete any of the above
LIST	List the current settings.

In all cases the input is the same. You are required to provide:

- A load-curve defining variation w.r.t. time for the transient analysis;
- A scale factor on the loads
- An optional load-curve for the dynamic relaxation phase (default none)
- For the velocity cases only a center of rotation

Removing applied base loads using ***DELETE***

Rather than applying null loads or load-curves use the ***DELETE*** option to kill loading in the selected mode(s).

Note:

- "Base" loads generate nodal forces, therefore their components are independent and the effect is additive to any other forces acting on nodes.

5.7 ***SPRINGS*** : Processing of Springs and Dampers

Summary: Spring and Damper elements in DYNA3D have special attributes and material definitions that make them different to "conventional" solids, shells and beams. This section deals with them.

In version 3.4A of LS-INGRID this menu is incomplete.

The ***SPRINGS*** menu is:

CREATE	[Not yet available]	Create springs and dampers
MODIFY	[Not yet available]	Modify springs and dampers
DELETE	[Not yet available]	Delete springs and dampers
LIST	[Not yet available]	List selected spring and damper status
STATUS		Summarize status of all springs to graphics area.

At present this menu may only be used to list a summary of all springs and dampers to the screen. This lists:

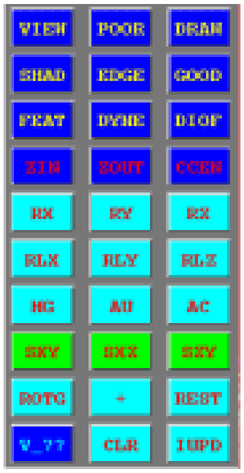
- Topology
- Type (Spring, Damper, etc.)
- Mode (Translational or Rotational)
- Material number and type
- Scale factor
- Orientation vector
- Print flag

6.0 USING DATA VISUALISATION MAIN MENU COMMANDS

This section describes the use of the data visualization (as opposed to processing) commands in the main menu referred to in section 4.3.

6.1 The Viewing Control Panel

The main viewing and plotting commands in LS-INGRID are contained in the viewing control panel located at the lower right of the screen. The layout of this is:



The Viewing Panel

The commands in the viewing panel are always available ("globally active") anywhere in LS-INGRID where plotting may take place. Therefore you can always redraw or modify your view regardless of what you are doing.

The commands mean the the following:

Commands that generate "wireframe" plots with no hidden-surface removal

DRAW Draws all vectors. All back and internal faces of solids are drawn, no hidden-surface removal occurs.

POOR Draws all vectors, but internal and back-facing faces of solids are removed. For models with solid elements this will produce a less cluttered plot than ***DRAW***. (The name is from "poor man's hidden-surface algorithm.)

Commands that generate "hidden-line" plots with hidden-surface removal

GOOD On 2D devices uses the painter algorithm which is quick, but can make mistakes where facets cross or there are facets of very different sizes close to one another.

On 3D devices will invoke the hardware hidden-surface removal algorithms, which will be Z-buffered.

VIEW On 2D devices uses a Z-buffered scan-line algorithm which is slow, but which will render crossed facets correctly.

On 3D devices this has no meaning.

Commands that generate "shaded" plots using a lighting model

SHAD On 2D devices uses the "painter" algorithm together with a lighting model to render a flat-shaded polygonal image with a hidden-line overlay.

On 3D devices uses the hardware to achieve a shaded image with hidden-line overlay.

S24 Is not yet available.

Commands that rotate the current view

RX RY RZ Provide rotation about the screen X, Y and Z axes respectively. This "screen" axis system is a right-handed cartesian system with its origin in the center of the screen: X being to the right, Y vertically upwards, Z out of the screen towards the viewer.

RLX RLY RLZ Provide rotation about the model X, Y and Z axes respectively. This is initially coincident with the screen system, but can rotate with respect to it. The "triad" in the lower left region of the graphics area shows its current orientation.

Commands that change the current image center and scale

ZIN ZOUT Zoom in and out respectively. To zoom in use the cursor to pick the diagonally opposite corners of the rectangular screen area that is to be enlarged to fill the screen. To zoom out pick the diagonally opposite points of a rectangular screen area that the current screen image is to be reduced to fill.

MG Magnify the scale of the current image by a factor. Values less that 1.0 make it smaller, values greater than 1.0 larger. The current center is not changed.

AU AC Autoscale the Unmodified geometry and the Current geometry respectively. ***AU*** will reset the image center and scale so that the full unmodified geometry would fit neatly on the screen in the current orientation. ***AC*** resets center an scale so that the "current" geometry, as blanked etc, fits neatly on the screen in the current orientation.

CCEN Uses the cursor to pick the point that is to be moved to the center of the screen. (i.e. a translation in screen X and Y) The current scale is not changed.

U/D L/R Move the image Up and Down, and Left and Right respectively. So it is translated in screen X and Y, but not scaled.

Commands providing preset views

SXY Rotates the image to give a plan view on the model XY plane. The scale is not changed.

SXZ Rotates to give a plan on model XZ plane, scale is unchanged1.

SZY Rotates to give a plan on model YZ plane, scale is unchanged.

REST Restores all aspects of the view to their current state, i.e.:

- Plan on Model XY plane
- Unmodified geometry autoscaled to fit screen

Other View Control commands

CLEAR Clears the graphics area of the screen completely. The current view settings are unchanged.

IUPD Toggles the "instant update" switch on or off (default off). If this switch is ON the image is updated immediately any of the viewing commands above are given; if OFF you must issue an explicit drawing command for the changes to become apparent.

V_?? Provides on-line help about all the commands in the viewing box. To use it first click on ***V_??***, then on the command on which you want help.

Using the slider bars for numeric input view commands

Those commands that require a numeric input: rotation, scaling and translation, will put up a slider bar when invoked.

Holding down the mouse button move it into this box and move the slider to the desired position, then release it.

To type in a value instead release the mouse button and type in a value at the dialogue prompt instead.

To abort a slider command hold the mouse down and move it out of the slider area, and it will disappear again.

Using "dynamic viewing" on 3D devices

On devices that support 3D graphics it is possible to use "dynamic viewing" as well as the explicit commands above.

This means using a combination of keyboard and mouse buttons to change the image rotation and scale in real-time. This is described in section 3.4.

6.2 ***LIGHTING*** : Control the lighting model used for Shaded plots

The lighting model consists of:

Ambient light	"Black body", non-directional light coming from everywhere that provides uniform illumination.
Directional light	Up to ten lights which can be positioned anywhere in space about the model. These provide directional lighting effects.

By default there is one directional light located at the observer, and there is a low level of ambient light. If you issue a ***SHAD*** viewing command it will be apparent that facets facing the observer are brightly lit, whereas those facing away from the observer get darker as the angle increases.

The ***LIGHTING*** menu is as follows:

CREATE	Lets you create up to 10 positional light sources;
MODIFY	Lets you modify the position and brightness of light sources;
SWITCH	Switches current light sources on/off;
DELETE	Deletes a light source definition;
AMBIENT	Controls the level of ambient light;
TWOSIDE	Controls two-sided lighting;
OBJECT	Sets the surface lighting attributes of the model;
LIST	Lists attributes of selected light sources;
STATUS	Summary of lighting model settings;
CMAP	Show current color map;
EXPLAIN	On-line help about lighting;
DETAILS	Detailed help on lighting model commands.

6.2.1 Setting ambient light level

The ambient light level comes from everywhere, so it has no position and provides no contrast. It has a possible range of 0 to 100% set by the ***AMBIENT*** command.

Typical values would be in the range 10% to 20%, the default being 15%.

To see how ambient light affects contract try setting it to 100% and then redrawing the image: all lighting effects disappear as the image is washed out.

Then set it to 0% and see how the dark areas become blacker.

6.2.2 Setting Positional light attributes

Positional lights exist in screen space. This is a right-handed space system that has its origin at the screen center, extending to +/- 0.5 in each of the screen X and Y dimensions, +ve Z being outwards.

Therefore a light over your right shoulder might be at screen space coordinates [3.0, 2.0, 5.0], and a distant light at the observer at [0.0, 0.0, 200.0].

When you ***CREATE*** or ***MODIFY*** a positional light you can define/modify:

- Its position in space, as described above;
- Its brightness in the range 0 - 100%.

When a light is created or modified you will be presented with a menu or predefined positions ("Observer", "Above", etc). You can select these or type in your own position.

By default a newly created light is switched on, but you can ***SWITCH*** it on or off at will without destroying its definition.

The light source is only truly killed when you ***DELETE*** it.

6.2.3 Setting object attributes

Some materials are matt, and some are shiny - both effects being independent of the lighting model. The ***OBJECT*** command lets you set the following attributes:

DIFFUSE The amount of diffuse light the object reflects: 0 - 100%. This will appear to make the object matt, and not shiny.

SPECULAR The amount of white light that facets reflect from the light source to the observer's eye position: 0 - 100%.

SHININESS The shininess of the object. Shiny objects reflect small, bright highlights; less shiny objects reflect broader, less intense highlights. The value is an integer in the range 1 to 10 that is an exponent: higher values give shinier results.

DONE Stores changes and makes them current for the next plot.

Notes:

- Not all graphics drivers support all aspects of shading. Therefore some of the commands above may have no effect on some hardware.

6.3 *GRAPHICS* : Control miscellaneous graphics attributes

This menu sets the following attributes of general graphics operations:

RESOL Sets the pixel (X,Y) resolution to be used for software hidden-line and shaded plots. The default is 1024 x 1024 pixels, but values in the range 256x256 to 4096x4096 may be set. There is no particular advantage on most screens on going to high resolutions, although lower ones will give faster plots.

TRIAD Toggles on and off the display of the model coordinate system triad at the bottom left corner of the plot.

TRPT Is a specialist command related to data plotting.

SHRINK Sets whether "element shrink" is on or not and, if on, what the shrink percentage is. Shrunk elements have their visible borders shrunk towards the element center making it easier to spot holes in the mesh.

MCOL Color mesh by Material Color; } These three options
PCOL Color mesh by Part Color; } are mutually
SCOL Color mesh by System Color; } exclusive.

6.4 ***DI*** : Display Options, extra information on plots

The "Display Options" menu allows you to display a variety of extra data on the current image.

If one or more options are selected the basic mesh colors are set to white, and the data to be shown is superimposed on the basic mesh in color.

These options are slowly being superseded by the "sketch" commands in the various data processing parts of LS-INGRID, but where duplication occurs the ***DI*** command has been kept to maintain backwards compatibility.

6.4.1 Using Display options

In normal usage a selection is made from the menu below, and this becomes the current display option superimposed on the plot. If a subsequent Display Option is selected it cancels the first one, i.e. only one option is current at a time.

If ***DIAD*** is used, then selection of DIsplay Options becomes additive: each new option does not cancel its predecessors. In this way several pieces of data may be displayed on the mesh simultaneously.

To cancel DIsplay Options, and hence return the basic mesh to its normal color system, the ***DIOF*** command must be used to turn all options OFF explicitly.

Generally nodal data is displayed as circles around the relevant nodes, and element data by sketching the segments or elements in question.

Where relevant arrows are drawn to show outward normal or other directions.

6.4.2 BCs_and_RESTRAINTS

The options in this menu display boundary conditions and restraints. Selected nodes are circled on the existing mesh. Options are:

DX	Show X translational restraints}	
etc to		} See the ***BOUNDS*** options in section 5.6
RZ	Show Z rotational restraints. }	
SY	Show nodes on symmetry planes	

FSYM	Show nodes on symmetry planes with failure
NRB	SHow nodes on non-reflecting (silent) boundary planes
CSYM	Show nodes with cyclic symmetry restraints

6.4.3 STRUCTURAL_LOADS

This sub-menu shows the nodes and elements subject to the following structural loads:

F	Nodal forces (point loads)
D	Nodes with prescribed displacements } (See ***BOUNDS***, section 5.6)
VB	Nodes with prescribed velocities }
CNV	Nodes on control volumes
PL	Edge pressure loads on elements
PR	Surface pressure loads on elements
DETP	Detonation point data

6.4.4 THERMAL_LOADS

This sub-menu shows the nodes and elements subject to the following thermal loads:

TI	Nodes with initial temperatures
TB	Nodes with temperature boundary conditions
FL	Elements with edge flux loads
FLUX	Elements with surface flux loads
CVL	Elements with edge convection loads
CV	Elements with surface convection loads
RBL	Elements with edge radiation loads
RB	Elements with surface radiation loads
REL	Elements with edge radiation enclosures
RE	Elements with surface radiation enclosures

6.4.5 GRAPHICS

This provides some extra graphics options:

MAT	Highlights a given material in color
PART	Highlights a given part in color
MK	Highlights a marked item
OUTL	Highlights shell free edges <--- **very useful !!**
LSRC	Draws light sources on the mesh

6.4.6 OUTPUT_AND_CONTROL

This sub-menu shows the nodes and elements included in the following output and control operations:

NPB	Nodes in nodal printout (time history) blocks
EPB	Elements in element printout (time history) blocks
CSEC	Nodes on cross-section force calculation planes
SPC	Nodes in SPC reaction groups
NFG	Nodes in nodal force groups
JOY	Nodes on a JOY interface
EDR	Elements to be deleted on restart

6.4.7 ENTITY_DISPLAY

This sub-menu shows structural items not normally drawn during plotting:

SPD	Displays springs and dampers	(Now included in normal plots)
JTS	Displays joints	(ditto)
SW	Displays nodes on stonewalls and walls themselves	
PM	Displays nodes of point (lumped) masses	
NV	Displays outward normal vectors of shell elements	
LAX	Displays local axes	
ORV	Displays orientation vectors	
BELT	Displays seat-belts etc.	(Now included in normal plots)
SYSJ	Displays system assembly joints	
TRAC	Displays tracer particles	
MPC	Displays multi-point constraints	
LSYS	Displays local coordinate systems	

6.4.8 INTERFACES

This sub-menu shows data related to interfaces and contact surfaces:

CS	Displays nodes & shells on selected surface (See ***SURFACE***, section 5.4)
TBS	Plots tie-breaking shell join lines
SL1D	Plots one-dimensional slide-lines
SBI	Plots shell to brick interfaces
INTF	Plots interfaces for component analysis

GEOM Plots geometric contact entities

6.4.9 SURFACES_AND_LINES

This sub-menu shows geometric surface and line data:

DS	Displays digitized surfaces
DSAD	Add further digitized surfaces to display list
DSRM	Remove digitized surfaces from display
L3D	Displays three-dimensional lines
PV	Displays tool paths
NSF	Displays NURBS surfaces
NCV	Displays NURBS curves
VDA	Displays VDA surfaces

6.4.10 DATA_ON_PLOTS

This sub-menu displays post-processing data in plots. This is a specialized process:

VECT	Displays data vectors
CONT	Displays data contours
FRIN	Displays fringe contours
WARP	Displays deformed geometry

6.4.11 *DIAD* and *DIOFF* : Setting and unsetting multiple DIsplay Options

Normally only one DIsplay Option is current, setting a new one cancels its predecessors. However:

DIAD Puts it into "additive" mode. Each new option is added to the current list and displayed with the others. This button may be toggled on/off to swap in and out of "additive" mode.

Finally, to turn DIsplay Options off and restore normal displays to their correct colors, use:

DIOF Turns all current DIsplay Options off again.

6.5 ***BLANK*** : Blanking and unblanking entities for display

The main viewing commands in LS-INGRID display the following entities:

3D elements	Solids and Thick shells;
2D elements	Thin shells and Contact surface facets;
1D elements	Beams, springs, dampers, seat-belts
Other elements	Joints, slip-rings, retractors, sensors, pre-tensioners and accelerometers.

By default all such elements that exist in the model will be drawn in the current mode. This can make displays cluttered and hard to understand, so it is possible to remove items from the display list by "blanking" them using this menu.

6.5.1 What "blanking" does

Items which are blanked are not deleted, they are simply not drawn. They can be "unblanked" again at any time to make them visible again.

Their current blanking status will not affect their visibility when drawn by other commands, i.e. ***DI***splay Options or the various ***SKETCH*** commands in data processing sub-menus.

6.5.2 The ***BLANK*** menu

BLANK	Sets function to "blank" mode } These are mutually exclusive:
UNBLANK	Sets function to "unblank" mode } selecting one unsets the other
STATUS	Lists current blanking status
ALL	(Un)blanks everything
PART	(Un)blanks everything in selected parts only
MAT	(Un)blanks elements of the selected materials only
SURFACE	(Un)blanks segments of the selected contact surfaces only
SOLID	(Un)blanks the selected solid elements
SHELL	(Un)blanks the selected thin shell elements
BEAM	(Un)blanks the selected beam elements
TSHELL	(Un)blanks the selected thick shell elements
SEGMENT	(Un)blanks the selected contact surface segments
SPRING	(Un)blanks the selected spring/damper elements

JOINT (Un)blanks the selected joint elements

BELT_etc (Un)blanks the selected seat-belt and related elements

6.5.3 Setting ***BLANK*** or ***UNBLANK*** mode

By default items selected will be "blanked", and you will see that the ***BLANK*** button in the menu is pre-selected. You may toggle between:

BLANK	Items selected are removed from the display: "blanked";
UNBLANK	Items selected are restored to the display: "unblanked";

These two modes are mutually exclusive, they unset each other.

6.5.4 (Un)blanking the whole model: ***ALL***

You can make the whole model visible or invisible by selecting ***ALL***. To see only a few items it is usually efficient to blank everything ("all"), then unblank just those items you want to see.

6.5.5 (Un)blanking by parts: ***PART***

You will be asked to select the parts to be processed. You can select them by name/number from the menu list, or by picking them from the screen using the ***VISIBLE*** options.

All items in the selected part(s) will be blanked or unblanked as appropriate.

6.5.6 (Un)blanking elements by material: ***MAT***

You will be asked to select the materials to be processed. You can select them by name/number from the menu list, or by picking them from the screen using the ***VISIBLE*** options.

"Materials" in this context refers only to the main DYNA3D materials used for solids, shells and beams; it does not include spring, seat-belt or other materials. All solid, shell and beam elements of the selected material(s) will be blanked or unblanked as appropriate.

6.5.7 (Un)blanking contact segments by surface: ***SURFACE***

You will be asked to select the contact surfaces to be processed. You can select them by number from the menu list, or by picking them from the screen using the ***VISIBLE*** options.

All segments of the selected surface(s) will be blanked or unblanked as appropriate.

6.5.8 (Un)blanking individual elements: ***SOLID, SHELL, BEAM, etc.***

In this context you are not given a menu of possible element numbers since the list would be too long. Instead you drop straight into the ***VISIBLE*** menu and you may screen-pick elements directly. You may, as always, type in explicit element labels instead if you wish.

All selected element(s) will be blanked or unblanked as appropriate.

6.5.9 (Un)blanking seat-belt entities: the ***BELT_etc*** sub-menu

Since seat-belts and their related entities are a specialist subject they are listed in their own sub-menu if they exist. This is:

ALL_TYPES	(Un)blanks all seat-belt and related types below
BELT_ELEM	(Un)blanks the selected seat-belt elements
SLIPRINGS	(Un)blanks the selected slip-ring elements
RETRACTORS	(Un)blanks the selected retractor elements
SENSORS	(Un)blanks the selected sensor elements
PRETENS	(Un)blanks the selected pre-tensioner elements
ACCEL	(Un)blanks the selected accelerometer elements

As with other element types these may be selected from the screen with the standard ***VISIBLE*** options, or have their labels typed in directly.

APPENDIX I

Structure of the Occupant "Tree" file

APPENDIX I Structure of the Occupant "Tree" file

1.0 Description

The "tree" file of a dummy is an optional, extra file that contains:

- The title and units system of the dummy;
- The dummy's "H-point" coordinates and local axis system;
- Topology and connectivity information about the dummy;
- Information about any contact surfaces pre-defined on the dummy.

If this file is defined for an occupant model then INGRID will be able to manipulate the dummy's sub-parts intelligently in an object-oriented fashion.

Throughout this Appendix the example used will be the "tree" file for the Oasys Rigid HYBRID III dummy. This is file:

```
/occupants/rh3/rh3.tree
```

This file is heavily commented and should, in itself, contain enough information to show you how to define such a file for your own dummy model.

Finally, "tree" files may be associated with any format of INGRID input file in which the material and node numbers are determinate: ie anything except a native "ingrid" format file. Moreover they could be used for any object-oriented system: it would not have to be a dummy model.

2.0 "Tree" filename syntax and location

The tree filename must be in the form:

```
<dummy acronym>.tree
```

It must be located in the same directory as the basic model **(.inf)** input deck of the dummy. Section 4.2.2 in the main manual describes how to define dummy pathnames, and how to load them into the code.

3.0 "Tree" file syntax

Numbers may be entered in "free" format, anywhere on a line. They should be separated by spaces or commas.

Character strings may be in upper or lower case, but note that part names must:

- Be in UPPER case;

- Be no more than 30 characters long, 15 or less is best;

- Be a single string with no gaps. (Use underscores to connect words, eg UPPER_TORSO, FOOT_TO_FLOOR);

Comment lines, starting with "c" may be inserted anywhere in the file; and blank lines may also be left anywhere.

4.0 Structure of the "tree" file

The contents of the file must be defined in the following order:

Title.

The title is assumed to be the first non-blank and non-comment line in the file. It may be up to 80 characters long, upper or lower case, and may include gaps.

Units system

You must define the <mass>, <length> and <time> units for your model. These may be defined in any order and must be one of the following consistent sets:

SI	:	***kg***	***m***	***s***
Modifed metric	:	***Te***	***mm***	***s***
Imperial	:	***lb***	***in***	***s***

H-Point coordinates

Define the [X,Y,Z] coordinates in global cartesian dummy model space of the dummy "H-Point" coordinates. Usually this will be [0,0,0], but you could offset this.

Local coordinate systems

You must define the dummy's local X and Z axes by means of vectors of unit length. It is suggested that you adopt:

X axis : Horizontal axis through shoulders;
Z axis : Vertical axis up through head.

For example the rigid RH3 dummy uses:

`0.0  1.0  0.0` Dummy local X axis is global Y;
`0.0  0.0  1.0` Dummy local Z axis is global Z.

Dummy rotations obey a right hand screw rule for rotations about these axes.

Define the number of dummy sub-parts

You may have up to 50 sub-parts in a dummy. A description of sub-parts is given lower down under "defining sub-parts".

Define the number of pre-defined contact surfaces

If your dummy has contact surfaces on it give the **total** number of them here (not just those you might use later), up to a maximum of 60.

Define the number of pre-defined contact surface groups

If you want to use contact surface groups give the number here, up to a maximum of 10.

You can assemble pre-defined contact surfaces into "groups". This concept is described in section 5.2.5, but briefly:

- Some aspects of contact may require more than one surface to fulfil a function. For example a belt contact might have to run over several unconnected materials, and this will require several distinct contact surfaces. Therefore it is convenient to be able to group the primitive contact surfaces together by function into "groups". Such a group may contain up to 10 primitive contact surfaces.

Defining sub-parts

Each sub-part is a collection of up to 60 materials, which can be rigid or deformable, and which need not be connected. For rigid materials "extra nodes on rigid bodies" are included in the sub-part definition.

INGRID treats each sub-part as indivisible, and transforms all nodes within it as if it were rigid.

The first, or "root" part is special:

- It is the starting point for the dummy object hierarchy: it has children but no parents.

- It rotates about the H-Point, and strictly should contain it.

Subsequent parts are children, or grand-(grand-)children of this root part.

Sub-parts are connected by coincident nodes, one on the parent part and one on the child. When sub-parts are rotated the dummy can be reassembled by translating sub-parts so that these coincident nodes (separated by rotation) become coincident again.

Sub-parts are defined by the following data, in this order:

Item	Data type
<name>	[Character string]
<initial angle>	Real number, in degrees
<load curve no> (See note 1 below)	Integer, zero if none
<#materials in part>	Integer, 1 - 60
<list of material nos>	Integer list
<#child sub-parts>	Integer, 0 - 5
<child sub-part, this node, child node>	3 integers

So a typical example of a sub-part entry might run:

```
EXAMPLE_PART_1                    (Part name)
10.5   0                          (Initial angle 10.5 deg, no load-curve)
2   13  20                        (Has two materials, nos: 13 and 20)
3                                 (Has 3 child sub-parts)
2  1345  7654                     (Sub-part #1 is 2, nodes 1345 and 7654)
6  1346  8966                     (Sub-part #2 is 6, nodes 1346 and 8966)
9  1356  9763                     (Sub-part #3 is 39 nodes 1356 and 9763)
```

These entries should be repeated for all sub-parts in the model.

Notes:

(1) Load-curves are an anachronism necessary before flexion-torsion joints became necessary in DYNA3D. They were used to define the stiffnesses of rotational springs around joints. Users creating new dummies should use flexion-torsion joints and set this value to zero.

(2) The first sub-part (#1) **must** be the "root" part. Thereafter the order does not matter, although parts should be defined in ascending order.

(3) Sub-parts must have only one path through the "tree" structure to the root sub-part. For example it is illegal to create a complete circle of sub-parts.

(4) The two coincident nodes used to link parts together **must** be given in the order <parent> <child>. They don't have to be nodes on joints, or any other element, but they must be associated with at least one material in the sub-part. They are often non-structural "extra nodes on rigid bodies".

(5) The remotest sub-parts (ie the youngest children) must implicitly have no children of their own. If they have you have made a mistake!

(6) It doesn't matter where you put <new lines> between numbers above. The layout here simply makes sense to the author.

Define contact surface names

For all the contact surfaces in the model define names. These should be upper case, 15 characters or less, with no embedded white space. Each name should start on a new line.

If you don't want a name use ***<none>*** or some other non-blank string.

Define contact surface groups

If you have any groups they must be defined as:

<Group name>   Up to 15 chars, upper case, no white space;

<Number of surfaces in group>   Integer, up to ten allowed.

<List of surface numbers>   List of integers

Therefore an example would be:

```
CONTACT_GROUP_1
4  13 14 20 30
```

(Has 4 surfaces: Nos 13, 14, 20 and 30)

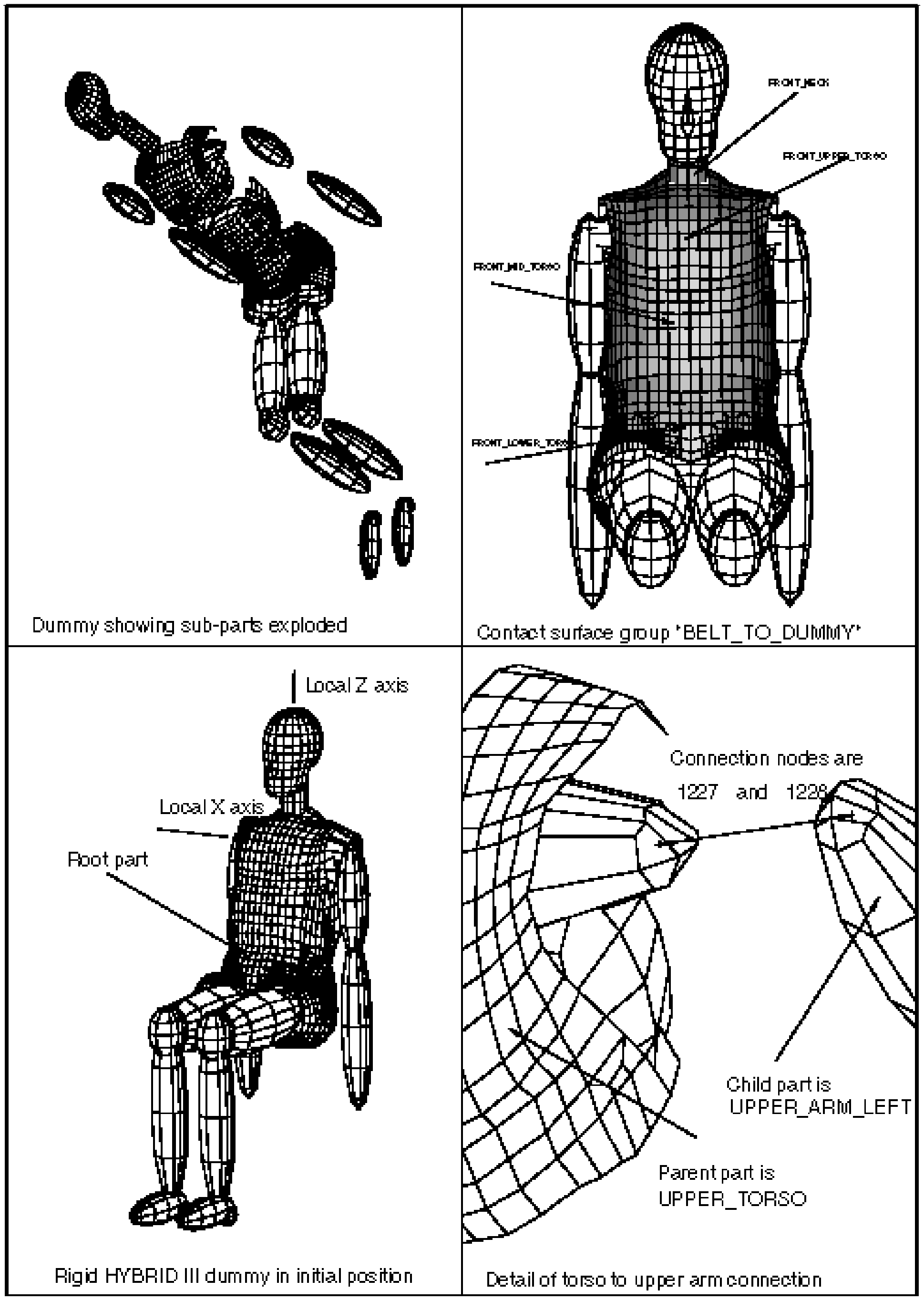
5.0 An example of a "tree" file

The full tree file for the Oasys Rigid HYBRID III dummy is reproduced below.

This is file:

```
/occupants/rh3/rh3.tree
```

Figure A/1 shows the arrangement of this dummy, an "exploded" view of its parts, a detail of the upper torso to arm connection, and one set of contact surfaces.



Dummy showing sub-parts exploded

Contact surface group "BELT_TO_DUMMY"

Rigid HYBRID III dummy in initial position

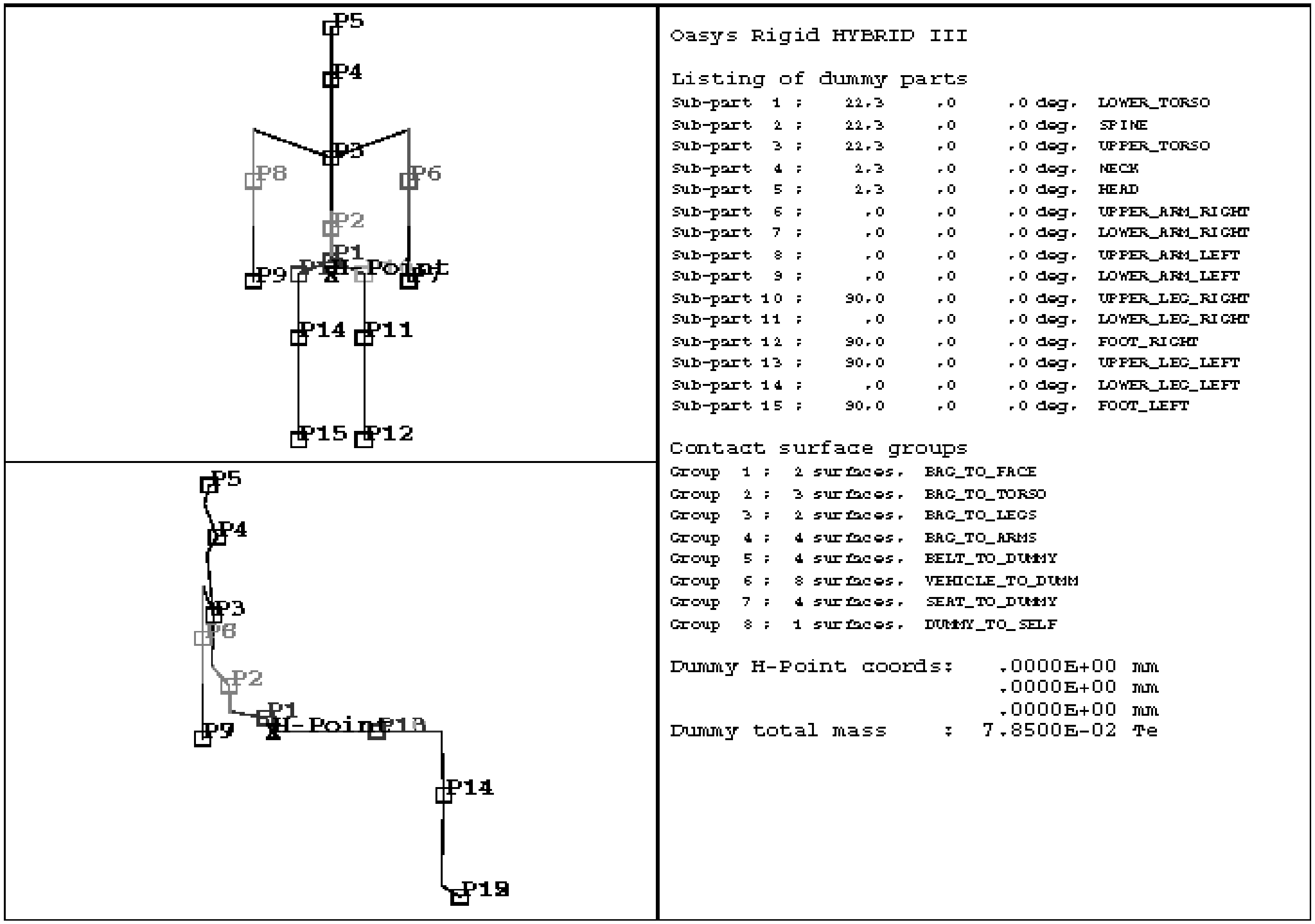
Detail of torso to upper arm connection

Appendix I: "Tree" file example using Oasys Rigid HYBRID III dummy

figure **A/1**

```
c Tree diagram file for Oasys Rigid hybrid III dummy.
c =============================================
c General format is:

c (1) Title (used as descriptor)
c (2) Units system used
c (3) H-Point coordinates
c (4) Rotation axis vector
c (5) Vertical axis vector
c (6) Number of separate parts (NPARTS)
c (7) Number of contact surfaces (NSURFS)
c (8) Number of contact surface groups (NSGRPS)

c For each part 1 to NPARTS:

c  ... Part name           (<= 15 characters preferred, 30 max, contiguous)
c  ... Initial orientation angle from vertical (degrees)
c  ... Load-curve for rotational spring stiffness that connects this part
c      to its master. Zero if none, -ve if orientation vector reversed.
c  ... Number of materials in part
c  ... List of material numbers in part
c  ... List of subordinate, connected parts

c      For each subordinate, connected part:

c      ... subordinate part number
c      ... connection node on this part
c      ... connection node on subordinate part


c For each contact surface  1 to NSURFS

c  ... Contact surface name


c For each contact surface 1 to NSGRPS

c  ... Surface group name  (<= 15 chars preferred, 30 max, contiguous)
c  ... Number of actual contact surfaces in group
c  ... List of actual contact surface numbers

c Comment lines may be inserted in the file using  "c "

c Input is free format for numbers and blank lines are ignored. INGRID
c pre-processor "calculator" logic would probably work, but I haven't
c tried it. There is no real need here.

c For character input:

c   ... The title is read as A80 format off the first non-blank/non-comment
c       line. Therefore it may contain separate words.
```

```
c   ... Names for parts, contacts, etc should be strings without white spaces.
c       Suggested convention is to use underscore (_) character between words
c       if you want to use multiple words in a title. They will be converted
c       to upper case internally.

c Title
Oasys Rigid HYBRID III

c Units system (modified metric)
   mm    Te    s

c H-Point coords
  0.0   0.0  0.0

c Initial rotation axis vector (Global Y axis)
  0.0   1.0  0.0

c Initial vertical axis vector (Global Z axis)
  0.0   0.0  1.0

c Number of parts (= no of materials for this model)
  15

c Number of contact surfaces
  28

c Number of contact surface groups
  8

c Part 1. Lower torso. Has 1 material, 3 subordinates: spine, upper legs
  LOWER_TORSO
  22.34  0
  1  1
  3
  2  1207  1208
 10  1247  1248
 13  1262  1263

c Part 2. Spine. Has 1 material, 1 subordinate: upper torso.
  SPINE
  22.34  37
  1  2
  1
  3  1212  1213

c Part 3, Upper torso. Has 1 material, 3 subordinates: neck, upper arms.
  UPPER_TORSO
  22.34  39
  1  3
  3
  4  1217  1218
  6  1227  1228
  8  1237  1238

c Part 4, Neck. Has 1 material, 1 subordinate: head.
  NECK
  2.34  0
  1  4
  1
  5  1222  1223

c Part 5, Head. Has 1 material, no subordinates.
  HEAD
  2.34  43
  1  5
  0

c Part 6, Upper arm right. Has 1 material, 1 subordinate: lower arm right.
  UPPER_ARM_RIGHT
  0.0   10
  1  6
  1
  7  1232  1233

c Part 7, Lower arm right. Has 1 material, no subordinates.
  LOWER_ARM_RIGHT
  0.0   14
  1  7
  0

c Part 8, upper arm left. Has 1 material, 1 subordinate: lower arm left.
  UPPER_ARM_LEFT
  0.0   16
  1  8
  1
  9  1242  1243

c Part 9, lower arm left. Has 1 material, no subordinates.
  LOWER_ARM_LEFT
  0.0   20
  1  9
  0

c Part 10, upper leg right. Has 1 material, 1 subordinate: lower leg right.
  UPPER_LEG_RIGHT
 90.0    3
  1  10
  1
 11  1252 1253

c Part 11, lower leg right. Has 1 material, 1 subordinate: foot right.
  LOWER_LEG_RIGHT
   0.0   7
  1  11
  1
 12  1257  1258

c Part 12, foot right. Has 1 material, no subordinates.
  FOOT_RIGHT
 90.0   9
  1  12
  0

c Part 13, upper leg left. Has 1 material, 1 subordinate: lower leg left.
  UPPER_LEG_LEFT
 90.0   25
  1  13
  1
 14  1267  1268

c Part 14, lower leg left. Has 1 material, 1 subordinate: foot left.
  LOWER_LEG_LEFT
  0.0   27
  1  14
  1
 15  1272  1273

c Part 15, foot left. Has 1 materual, no subordinates.
  FOOT_LEFT
 90.0   30
  1  15
  0


c Contact surface names.

 FRONT_FACE
 FRONT_NECK
 FRONT_UPPER_TORSO_1
 FRONT_MID_TORSO_1
 FRONT_LOWER_TORSO_1
 UNDER_RIGHT_FOOT
 UNDER_LEFT_FOOT
 REAR_UPPER_TORSO
 REAR_MID_TORSO
 REAR_LOWER_TORSO
 BASE_LOWER_TORSO
 RIGHT_UNDER_THIGH
 LEFT_UNDER_THIGH
 RIGHT_KNEE_&_SHIN
 LEFT_KNEE_&_SHIN
 UPPER_ARM_RIGHT
 LOWER_ARM_RIGHT
 UPPER_ARM_LEFT
 LOWER_ARM_LEFT
 NECK_CIRCUM
 FRONT_UPPER_TORSO_2
 FRONT_MID_TORSO_2
 FRONT_LOWER_TORSO_2
 CHIN_TO_CHEST
 RIGHT_THIGH_CIRCUM
 LEFT_THIGH_CIRCUM
 RIGHT_HAND
 LEFT_HAND


c Contact surface groups. These are logical groupings of the primitive
c contact surfaces.

c Group 1: bag to face. Owns face and neck contacts.
  BAG_TO_FACE
  2  1  2

c Group 2: bag to torso. Owns torso_1 contacts.
  BAG_TO_TORSO
  3  3  4  5

c Group 3: bag to legs. Owns thighs circumferential.
  BAG_TO_LEGS
  2 25 26

c Group 4: bag to arms. Owns arms circumferential.
  BAG_TO_ARMS
  4 16 17 18 19

c Group 5: Belt to dummy. Owns torso_2 + neck
  BELT_TO_DUMMY
  4 20 21 22 23

c Group 6: Dummy to vehicle.
  VEHICLE_TO_DUMMY
  8  6  7 12 13 14 15 27 28

c Group 7: Dummy to seat
  SEAT_TO_DUMMY
  4  8  9 10 11

c Group 8: Dummy to self
  DUMMY_TO_SELF
  1 24

c End of file.
```



```
Oasys Rigid HYBRID III

Listing of dummy parts
Sub-part  1 :    22.3     .0     .0 deg.  LOWER_TORSO
Sub-part  2 :    22.3     .0     .0 deg.  SPINE
Sub-part  3 :    22.3     .0     .0 deg.  UPPER_TORSO
Sub-part  4 :     2.3     .0     .0 deg.  NECK
Sub-part  5 :     2.3     .0     .0 deg.  HEAD
Sub-part  6 :      .0     .0     .0 deg.  UPPER_ARM_RIGHT
Sub-part  7 :      .0     .0     .0 deg.  LOWER_ARM_RIGHT
Sub-part  8 :      .0     .0     .0 deg.  UPPER_ARM_LEFT
Sub-part  9 :      .0     .0     .0 deg.  LOWER_ARM_LEFT
Sub-part 10 :    90.0     .0     .0 deg.  UPPER_LEG_RIGHT
Sub-part 11 :      .0     .0     .0 deg.  LOWER_LEG_RIGHT
Sub-part 12 :    90.0     .0     .0 deg.  FOOT_RIGHT
Sub-part 13 :    90.0     .0     .0 deg.  UPPER_LEG_LEFT
Sub-part 14 :      .0     .0     .0 deg.  LOWER_LEG_LEFT
Sub-part 15 :    90.0     .0     .0 deg.  FOOT_LEFT

Contact surface groups
Group  1 :  2 surfaces.  BAG_TO_FACE
Group  2 :  3 surfaces.  BAG_TO_TORSO
Group  3 :  2 surfaces.  BAG_TO_LEGS
Group  4 :  4 surfaces.  BAG_TO_ARMS
Group  5 :  4 surfaces.  BELT_TO_DUMMY
Group  6 :  8 surfaces.  VEHICLE_TO_DUMM
Group  7 :  4 surfaces.  SEAT_TO_DUMMY
Group  8 :  1 surfaces.  DUMMY_TO_SELF

Dummy H-Point coords:    .0000E+00 mm
                         .0000E+00 mm
                         .0000E+00 mm
Dummy total mass    :  7.8500E-02 Te
```

Appendix I: Output from OCCUPANT "LIST" option for RH3 HYBRID III dummy showing sub-parts and tree structure

APPENDIX II

Worked Examples of Airbag folding

APPENDIX II Worked examples of Airbag Folding

1.0 Description

Airbags are ordinary meshes using enclosed volumes of shell elements. They are inflated during an analysis using control volumes.

They may be folded in INGRID using the airbag folding options described in section 5.1 of this manual.

This appendix gives worked examples of the folding of two airbags:

- A circular "pancake" driver's side bag; (See figure A/2)
- A more boxy "pillow" passenger side bag. (See figure A/3)

2.0 Input files

Standard example files are provided with LS-INGRID for typical driver and passenger side airbags. If you do not have these files please contact your LS_INGRID distributor to receive them.

They are used as the basis of the examples in this Appendix, and figures A/2 and A/3 show their general (unfolded) shape.

2.1 Permission to use these files

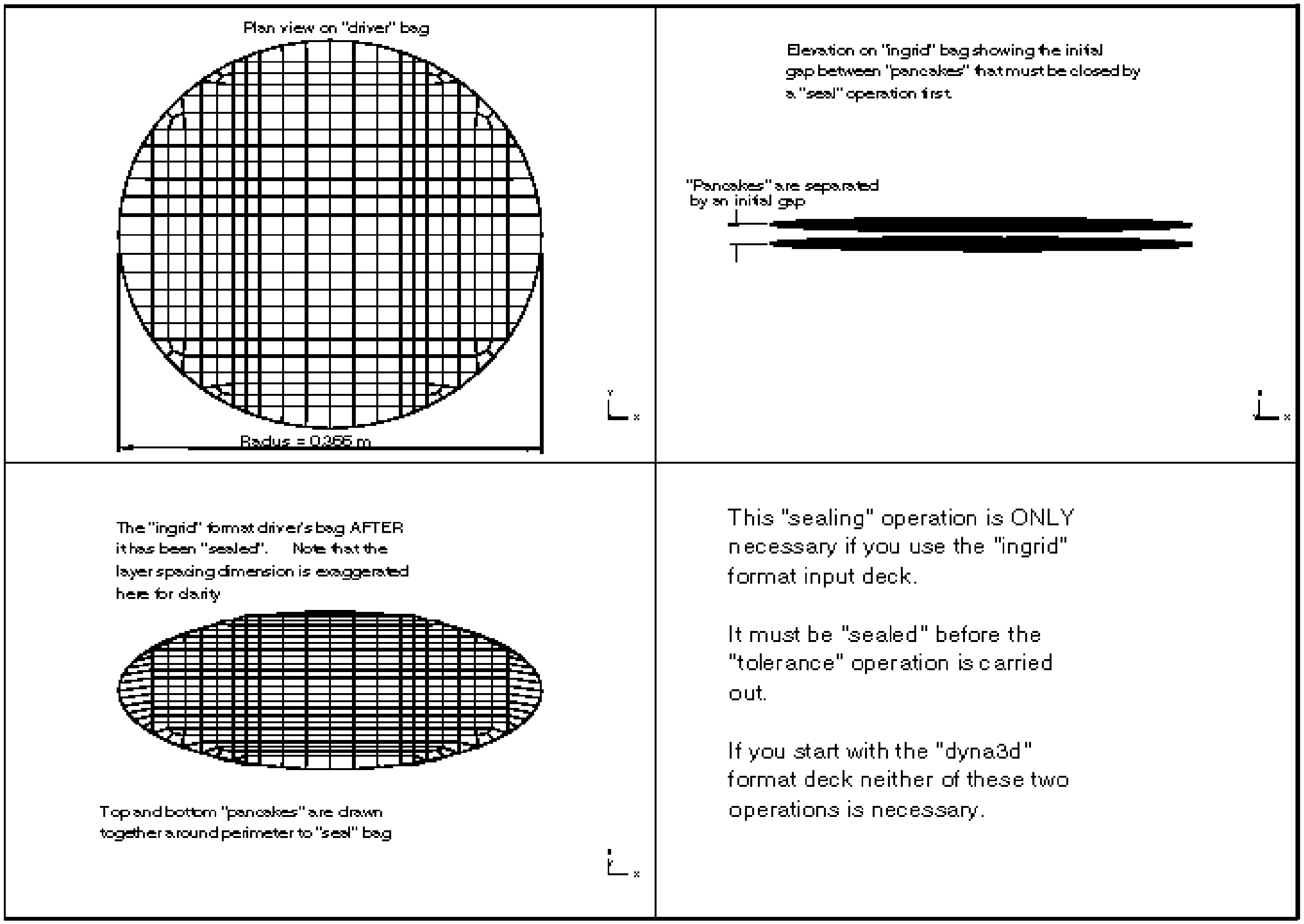
Permission is hereby granted for bona-fide customers of LS-INGRID to use, copy and modify these files at will.

2.2 Disclaimer

These models are intended as examples only.

Neither geometry, nor material properties, nor control volume characteristics, nor any other aspect of them is warranted to be correct.

Users must satisfy themselves that any models they build based on these are correct, and represent adequately the airbags they are trying to model.

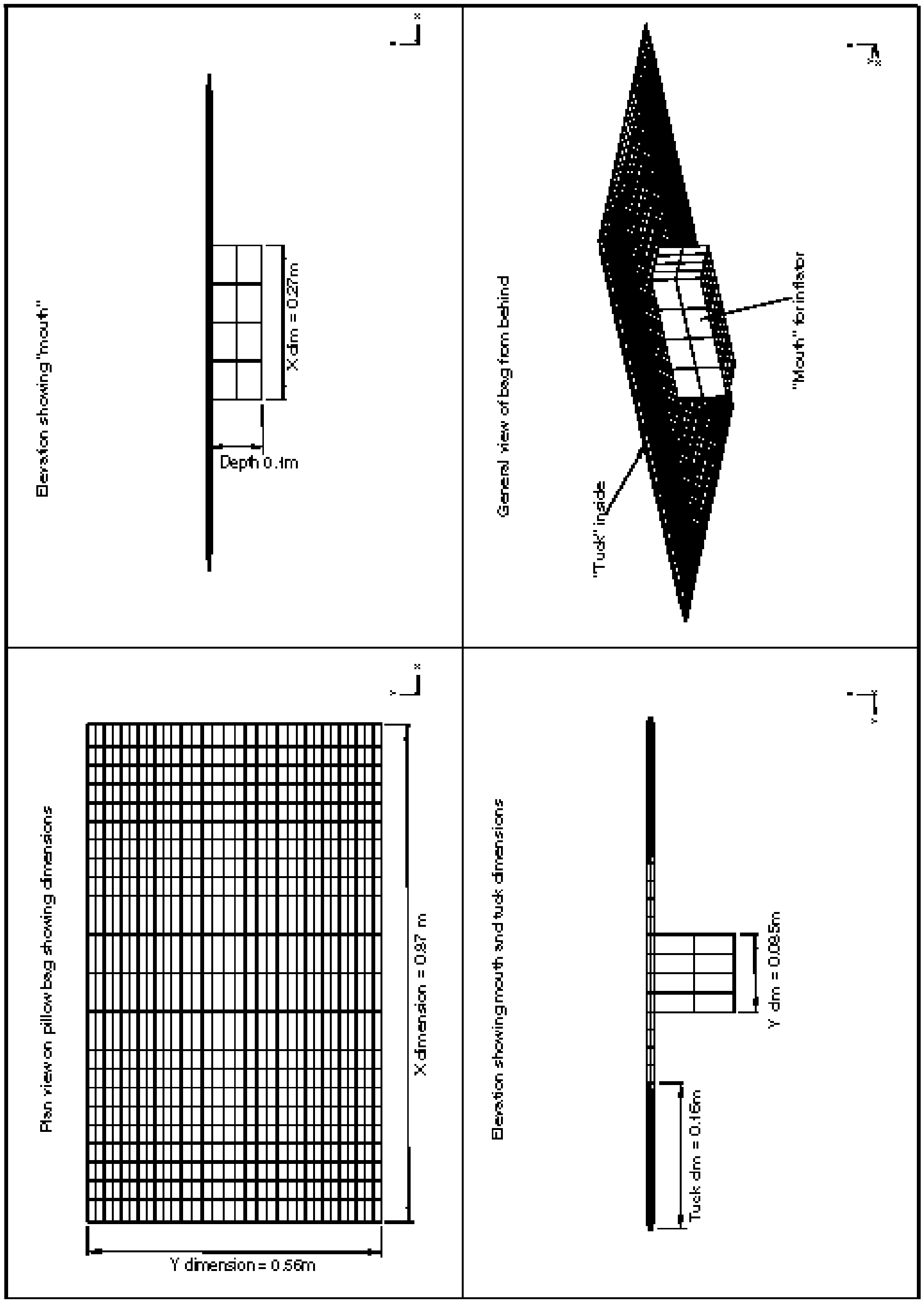

Plan view on "driver" bag
Radius = 0.366 m
Elevation on "ingrid" bag showing the inital gap between "pancakes" that must be closed by a "seal" operation first.
"Pancakes" are separated by an inital gap
The "ingrid" format driver's bag AFTER it has been "sealed". Note that the layer spacing dimension is exaggerated here for clarity
Top and bottom "pancakes" are drawn together around perimeter to "seal" bag
y
x

This "sealing" operation is ONLY necessary if you use the "ingrid" format input deck.

It must be "sealed" before the "tolerance" operation is carried out.

If you start with the "dyna3d" format deck neither of these two operations is necessary.

Appendix II: General layout of "driver" driver's side bag showing "sealing" needed in "ingrid" format decks figure **A/2**


Elevation showing "mouth"
X dim = 0.27m
Depth 0.1m
x
General view of bag from behind
"Mouth" for inflator
"Tuck" inside
x
Plan view on pillow bag showing dimensions
X dimension = 0.97 m
Y dimension = 0.56m
x
y
Elevation showing mouth and tuck dimensions
Y dim = 0.085m
Tuck dim = 0.16m
y

Appendix II: General layout of "pillow" passenger side airbag

figure A/3

2.3 "Ingrid" format files

Both of these meshes were originally generated using the "ingrid" input files:

```
/examples/driver.ing
/examples/pillow.ing
```

And users may wish to experiment with these files. They have been set up in a parameterised fashion so that their key dimensions and mesh densities may be adjusted.

2.3.1 Key parameters in the "ingrid" format driver's side file

Users may wish to adjust the following:

`[radius = 0.366]`	This is the bag external radius
`[mult   = 1]`	This is a mesh density multiplier

2.3.2 Key parameters in the "ingrid" format passenger's side file

Users may wish to adjust the following:

`[mult   = 2]`	Mesh density multiplier
`[xspan  = 0.87]`	Bag overall X dimension
`[yspan  = 0.56]`	Bag overall Y dimension
`[mouth_x = 0.270]`	"Mouth" X dimension
`[mouth_y = 0.085]`	"Mouth" Y dimension
`[mouth_z = -0.1]`	"Mouth" Z dimension (depth)
`[tuck    = 0.16]`	Size of tuck fold

2.3.3 IMPORTANT: Using the "ingrid" format files

When these files have been read into INGRID you **must** join their constituent parts together using the ***TOLERANCE*** command. A value of 0.1mm (ie 0.0001 in the SI units system used) is recommended.

For the driver's side bag only it will be necessary to "seal" the pancake edges together before this tolerance operation is carried out using ***AIRBAGS, SEAL***.

The reader is referred to the main INGRID manual for more explanation of these commands, and also to figure A/2.

2.4 "Dyna3d" format files

For users who do not wish to delve into "ingrid" format files these two models have also been generated as LS-DYNA3D version 930 formatted input decks.

These have been generated using the same mesh densities and dimensions as the "ingrid" format files. So figures A/2 and A/3 are equally appropriate. The files are:

```
/examples/driver.inf
/examples/pillow.inf
```

(You do **not** have to worry about ***SEAL*** and ***TOLERANCE*** operations if you start from these "dyna3d" format files: this has already been done for you.)

2.4.1 Pre-defined fold definitions for "dyna3D" format input decks

Pre-existing folds cannt be included in the DYNA3D input decks, so the pre-defined folds used in these examples have been written to the following fold definition "save" files:

```
/examples/driver_folds.sav
/examples/pillow_folds.sav
```

These are generic names: both an ASCII and a binary file are written when old definitions are saved. See section 2.5.3 below.

2.5 Folding the driver's side bag

You will observe from figure A/2 that the driver's bag has been meshed in such a way that orthogonal mesh lines are available in X and Y axes. These are the natural starting points for the mesh folding operation.

It is assumed that you are starting from the "dyna3d" format input deck:

```
/examples/driver.inf
```

Enter the ***AIRBAGS*** menu in the data processing part of the code, and...

2.5.1 Defining what to fold

By default the whole model is selected for folding. And in fact this is what is required here. But if this were part of a larger model (eg a car) you would have to select just the airbag material for folding. Do this by:

SELECT, REMOVE, ALL	To deselect everything
ADD, MAT, VISIBLE	And select the airbag material

Check that you have got this right:

SKETCH	To see what is selected

2.5.2 Defining folds

Clearly you may start where you wish and fold what you want, but remember that it is better (if possible) to work from the outside of the mesh inwards.

Before you start folding have a plan: figure A/4 shows the planned fold lines for this mesh. Note that folding is assumed to be symmetrical and only half the folds about each axis are shown.

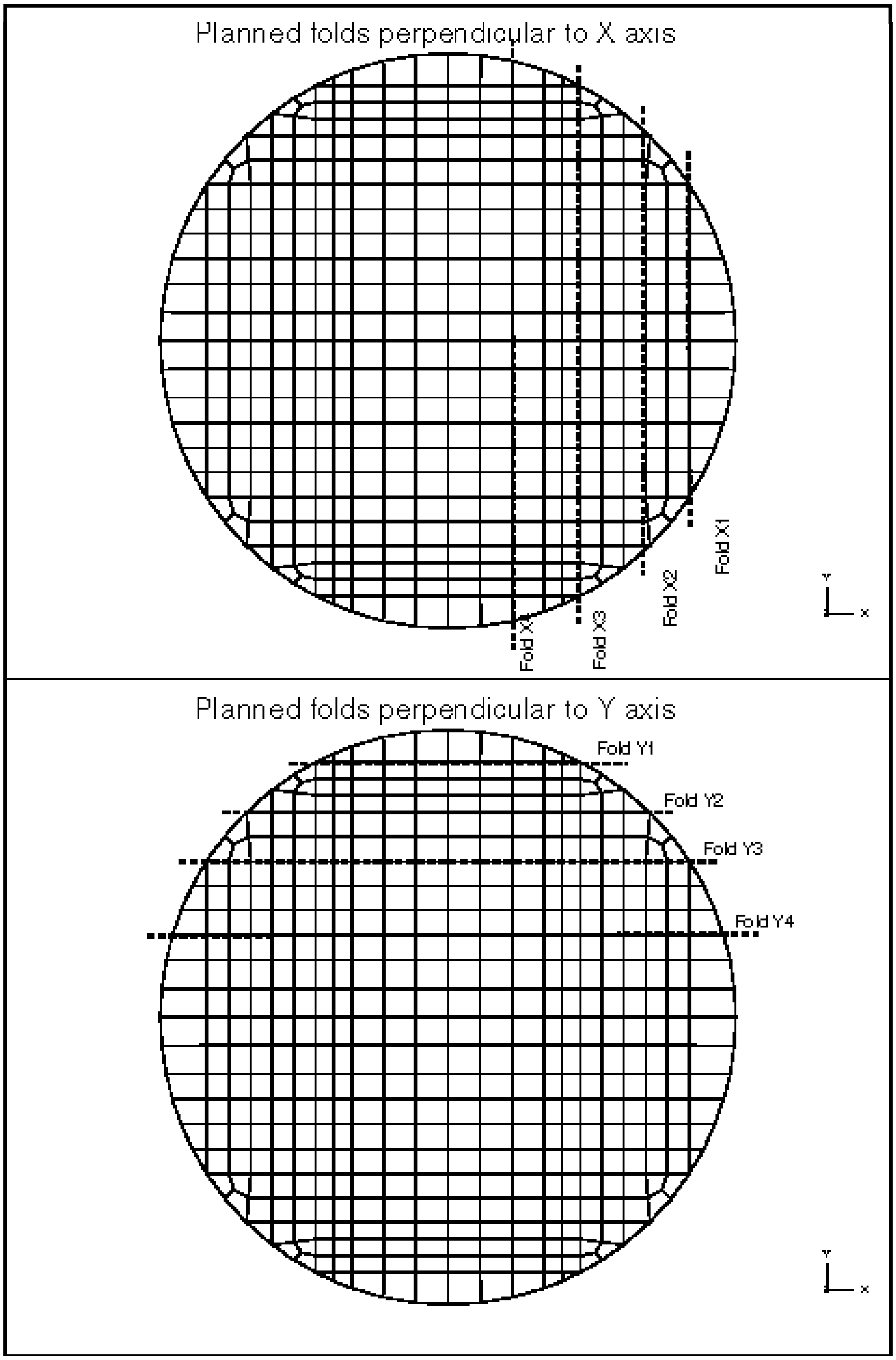
In this example we will make the first fold the mesh line perpendicular to the X axis that has the most +ve X coordinate. This is shown in figure A/4 as fold X1.

<u>Define the first fold by:</u>

DEFINE, NODE_NODE	To pick two nodes (see fig A/5)
+	To use +ve X side of line
UP	To fold it up over mesh
1.00E-3	To use fold thickness 1mm
1.0	To use a scale factor of 1.0
1.0	To use a radius of 1mm
THIN	To use a "thin" fold

At this point the fold will be sketched on the mesh as shown in figure A/5, and you will be asked if this fold definition is OK.

OK	Will confirm and store fold.


Planned folds perpendicular to X axis
Fold X
Fold X3
Fold X2
Fold X1
Y
x
Planned folds perpendicular to Y axis
Fold Y1
Fold Y2
Fold Y3
Fold Y4
Y
x

Driver's side airbag showing planned X and Y fold-lines figure A/4

Check this fold by:

```
CHECK                                   To check fold geometry.
```

This should return the message **"Folds check out OK"**.

Execute this fold by:

```
EXECUTE, ALL                  To make all folds active
         1.0e-3                          To choose a min thickness of 1mm
```

Check the fold visually

Use any viewing command (eg ***POOR***) to draw the newly folded shape. You should see the image shown in the lower half of figure A/5.

If you change view using ***SXZ***, and zoom in on the fold (***ZIN***), you will probably find it difficult to see what has happened (See top half of figure A/6). Therefore, to make visualisation easier, make the fold thicker by increasing the minimum thickness used for all folds, without changing the individual thickness of this fold. (Read 5.1.2(j) in this manual for more information.)

```
EXECUTE, ALL                  To make all folds active
         2.5e-3                          Increase min thickness to 2.5mm
```

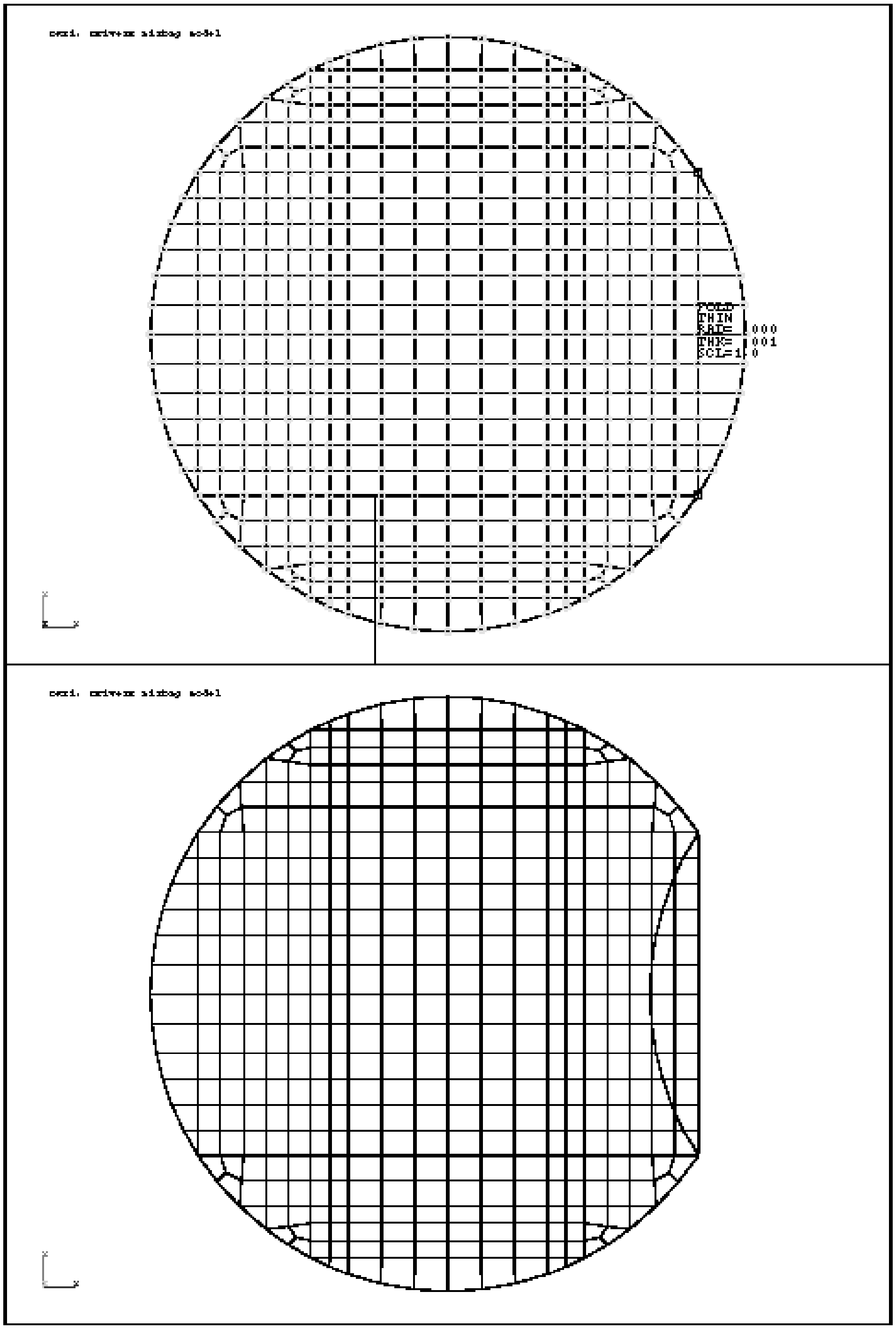
Now draw it again, and (zoomed in) you should see the thicker fold shown in the lower half of figure A/6.

Define the second fold (X2 on figure A/4)

For this fold it is clear that the mesh lines do not lie exactly on the fold line, so we must adjust the mesh. So get back to a plan view on the XY plane showing the whole bag:

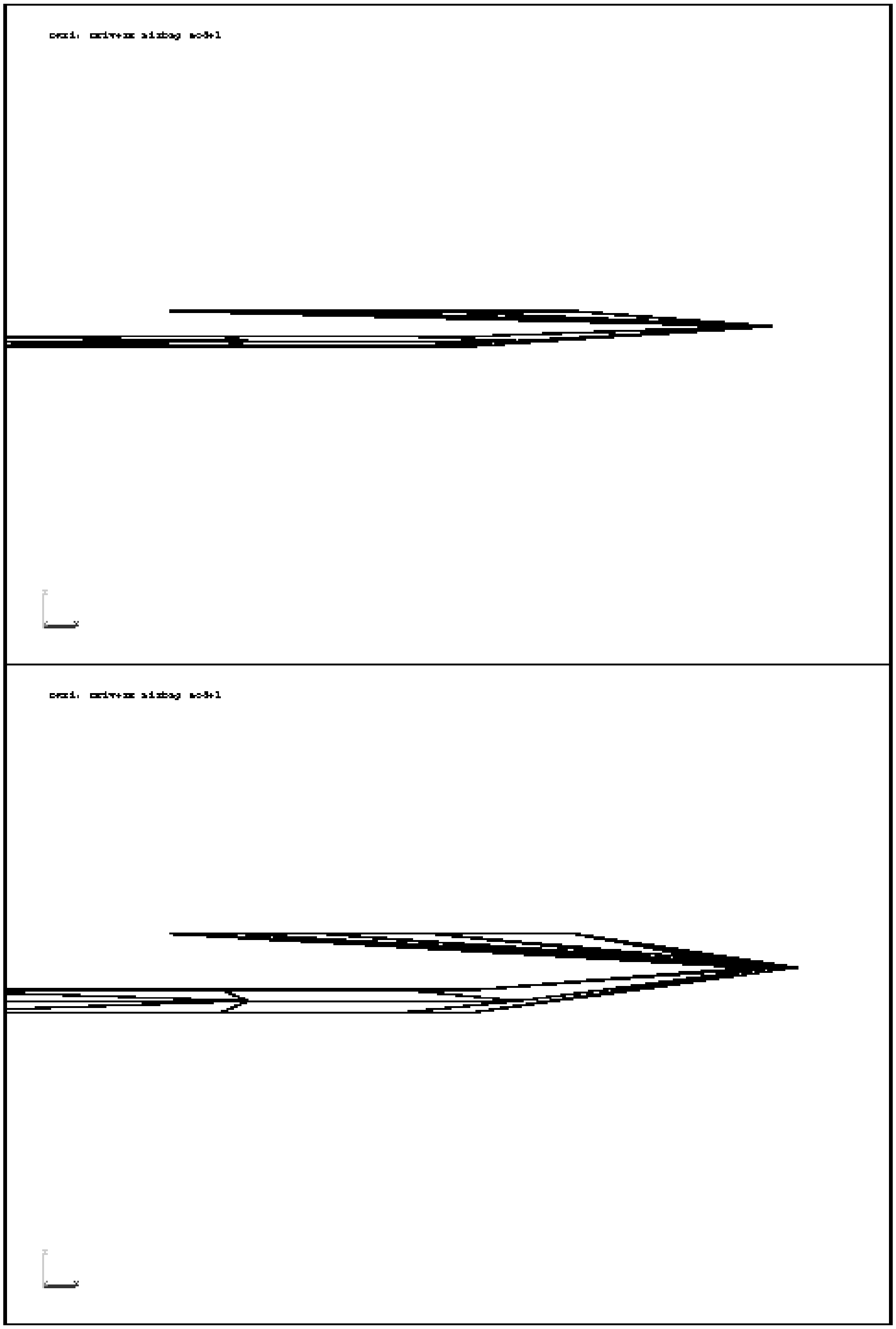
```
SXY, AC, POOR                           Plan on XY, autoscaled & redrawn
```

Then define the next fold as before by using two nodes. This time do not use the nodes on the outside of the mesh but those a bit further in as shown by the pick symbols in figure A/7:



Appendix II: Driver's side bag showing latent fold definition #1, and its new folded state

figure **A/5**



Appendix II: Driver's side bag showing section through fold #1 at 1mm and 2.5mm thicknesses

figure **A/6**

DEFINE, NODE_NODE	To pick two nodes (see fig A/7)
+	To use +ve X side of line
DOWN	To fold it down under mesh
1.00E-3	To use fold thickness 1mm
1.0	To use a scale factor of 1.0
1.0	To use a radius of 1mm
THIN	To use a "thin" fold

But now, when you check it, you will find that some nodes are flagged as not being on the fold line:

CHECK	To check folds

This time you will get the message **"Fold 2 has 4 nodes needing moving"**, and the four nodes (2 pairs on top of one another) will be circled. And you will get the prompt:

Move nodes?

The upper half of figure A/7 shows these nodes, and it is clear that not enough nodes have been detected by the "check" algorithm: the outermost nodes should be moved as well. Therefore you must choose the option:

CH_TOL, 8.0e-3	To change check tolerance to say 8mm

And the check will be repeated. This time it will be wide enough to pick up 6 nodes, and you will see from the lower half of figure A/7 that the outer diameter nodes have been picked this time. So now you can reply:

OK	To agree to move the nodes.

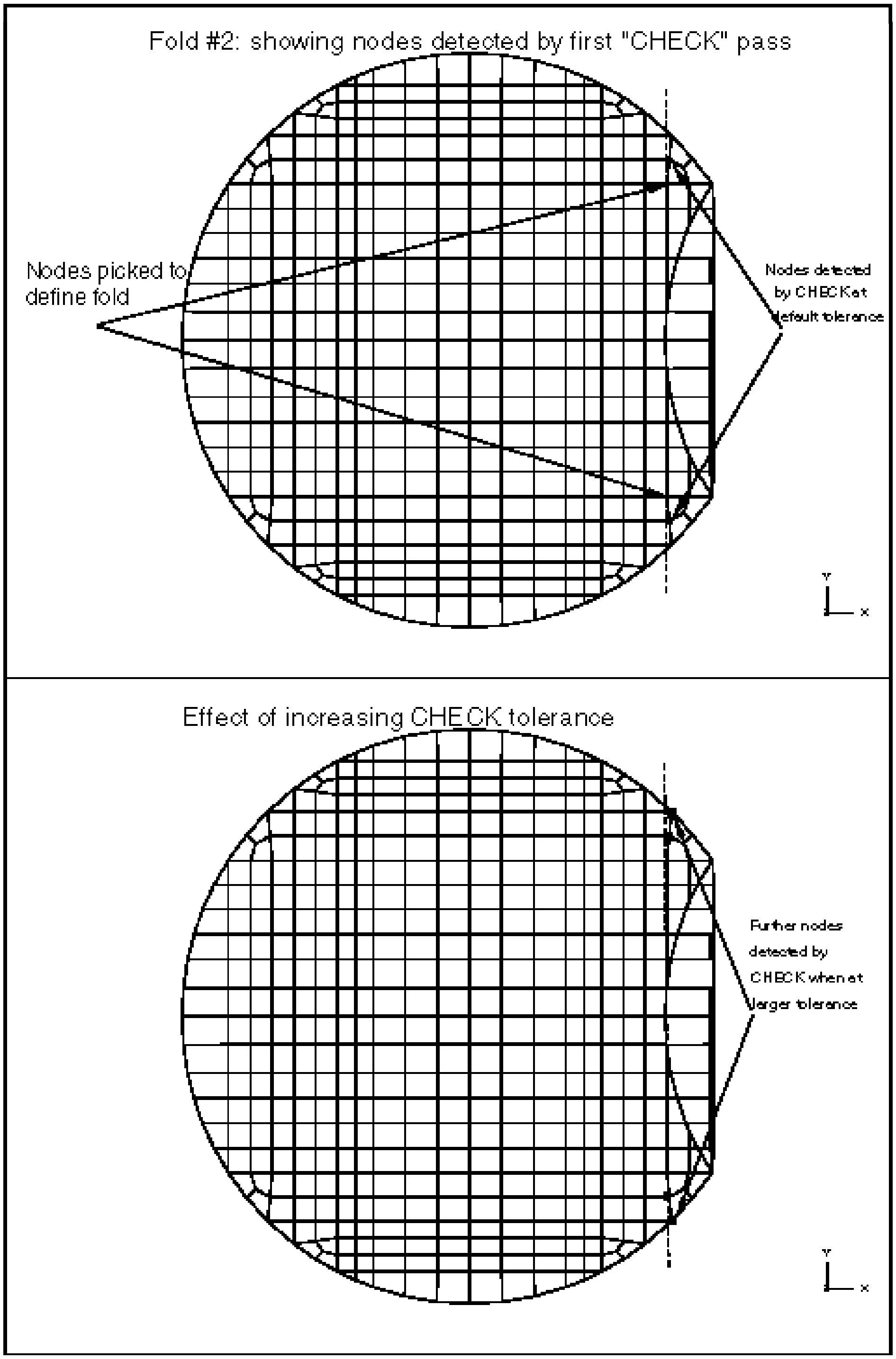
You will be told that 6 nodes have been moved, and warned that the airbag has been unfolded. To see what the effect of moving these nodes was type:

EXECUTE, 1, 2.5e-3	To re-execute fold 1 only

And redraw. You will see that the bag coordinates have been adjusted slightly, but that the nodes now lie on fold #2. This causes a small mesh distortion, see figure A/8 (upper half), but it is acceptable.

If you now execute fold 2 as well by:

EXECUTE, ALL, 2.5e-3	Re-execute all folds



Fold #2: showing nodes detected by first "CHECK" pass

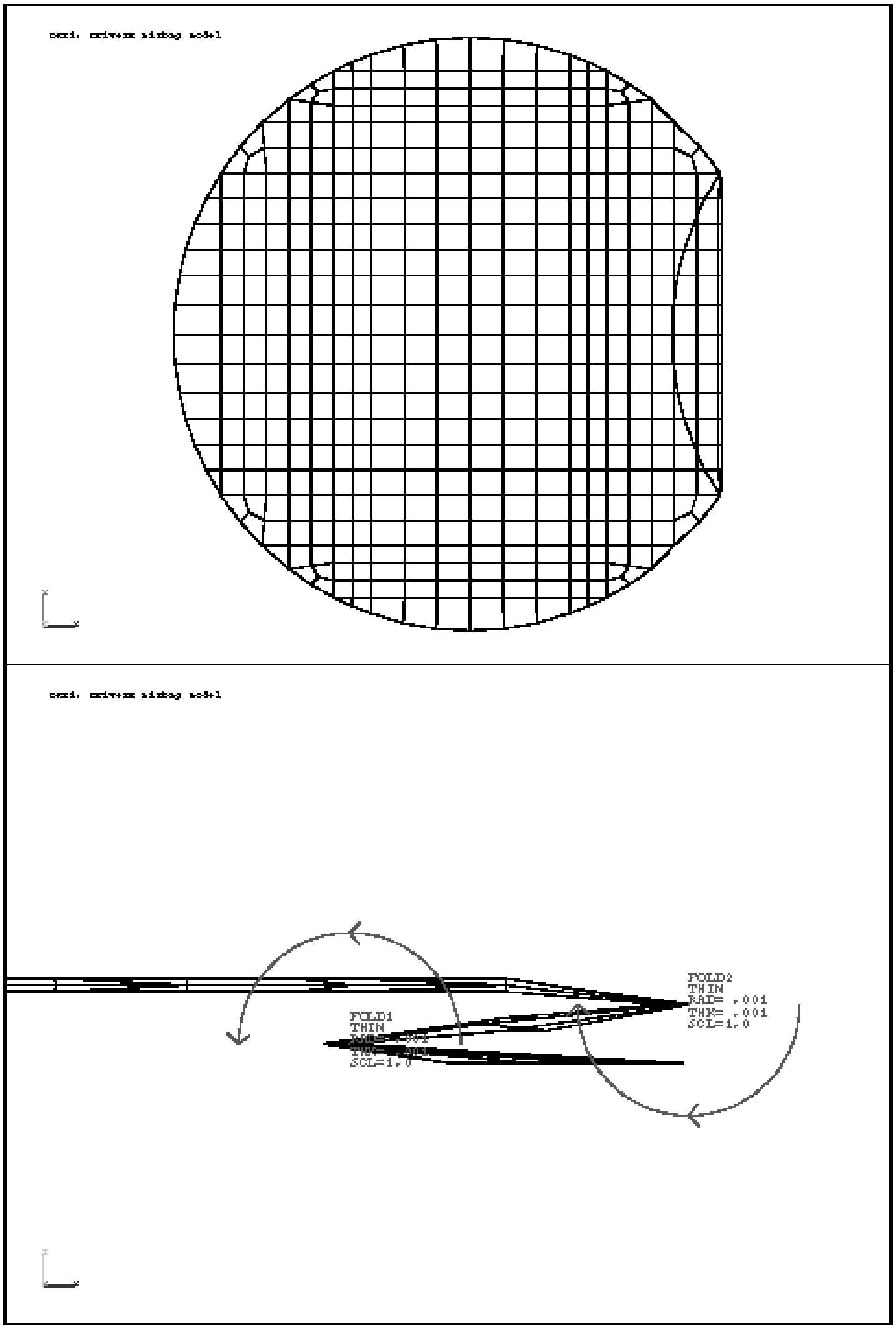
Nodes picked to define fold

Nodes detected by CHECK at default tolerance

Effect of increasing CHECK tolerance

Further nodes detected by CHECK when at larger tolerance

Appendix II: Showing how to adjust nodes on fold #2 figure **A/7**



Appendix II: Driver's bag. Effect of adjusting mesh using CHECK, and resulting folds #1 and #2

figure **A/8**

You will now see both folds. If you zoom in on a section through the folds, and then "sketch" all folds (using the ***SKFALL*** button in the top box), you will see how the two folds are now arranged. The lower half of figure A/8 shows such a section.

To save you going through the ***EXECUTE, ALL, <thickness>*** command sequence each time you re-execute folds you can use the short cut of ***REFOLD*** in the button box as the top of the screen. This executes that command sequence for you.

Execute the remaining X folds

The same process can be carried out for folds X3 and X4. When you get to fold X4 you will find some more nodes that need moving slightly, so use ***CHECK*** as before to adjust them.

You will also find a different problem: fold X4 will, if the up/down/up/down sequence is followed, end up putting the bag underneath instead of on top of the mesh. See the top half of figure A/9.

So you will have to go through folds X1 to X4 changing all ***UP*** flags to ***DOWN***, and vice-versa. Do this as follows:

```
MODIFY, FOLD_1, V5      To change fold #1 up/down flag
                -1      To change it to "down"
                SAVE          To save changes
```

Repeat for folds #2 to #4 inverting the up/down flag in each case, then re-execute the folds and redraw:

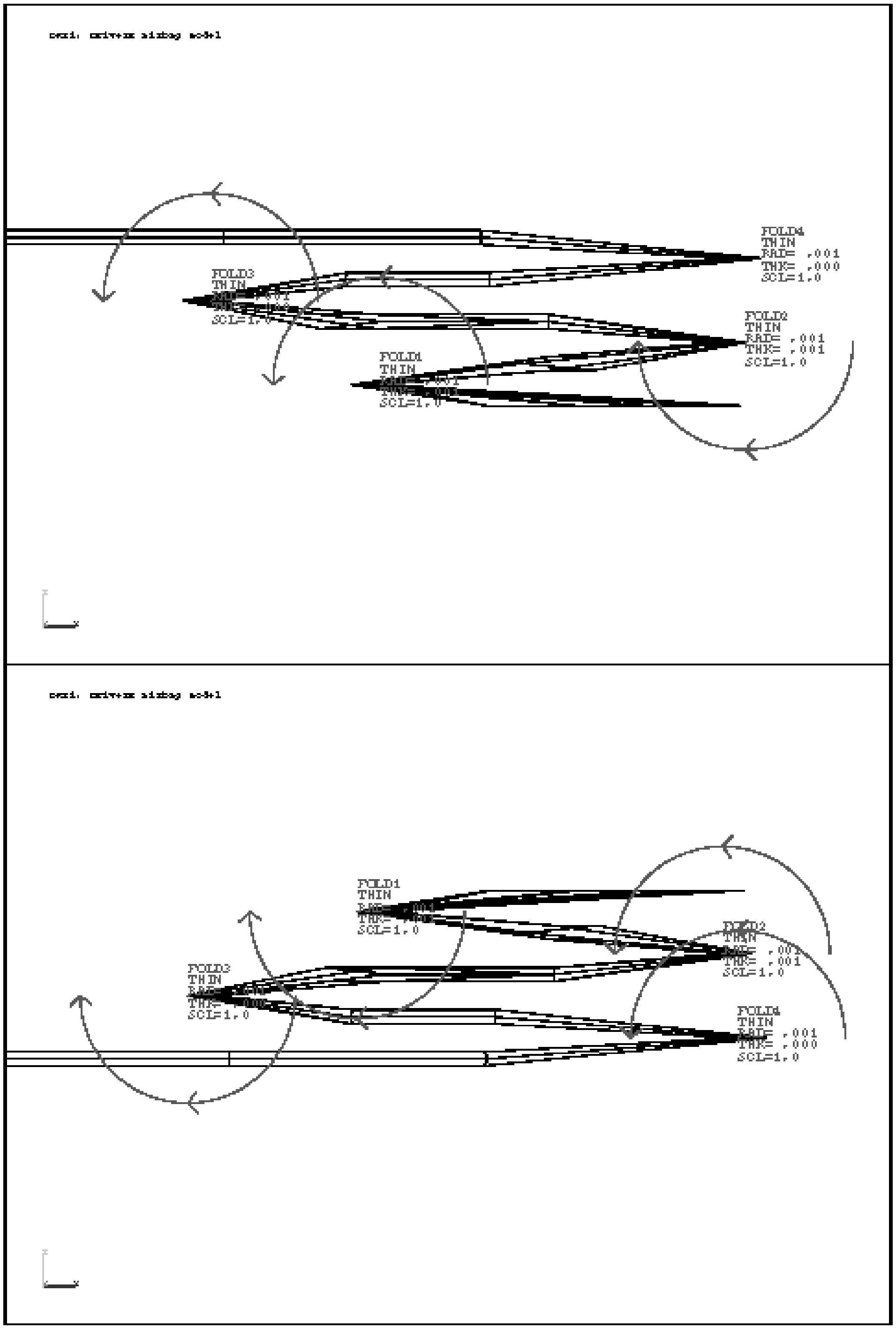
```
REFOLD                  To re-execute folds
SKFALL                  To re-sketch fold definitions
```

This will produce the revised fold pattern shown on the bottom half of figure A/9.

Copy X folds #1 to #4 to opposite side of bag

We need to copy folds #1 to #4 anti-symmetrically about the mesh centreline to form folds #5 to #8. This is simple:

```
COPY, ALL                     To copy all existing folds
      REFLECT                 To copy anti-symmetrically
      0.0                     0.0 is centre of bag
```



Appendix II: Driver's bag. Initial layout of folds
#1 to #4, and the results of reversing up/down flags

figure **A/9**

It is a good idea to check that the reflected folds produced this way are in the right place. This is most easily done on the unfolded geometry, so:

RESET	To unfold bag
SXY; AC	Autoscaled plan view on XY
SKFALL	To sketch all folds
CHECK	To check the new folds

This, not surprisingly, shows that the same nodes need to be adjusted on the opposite side of the mesh: see top half of figure A/10. So do that:

[CHECK], OK	To adjust nodal positions

Then refold the mesh, and have another look at a section through it:

REFOLD	To re-execute all folds
SXZ; ZIN	To zoom in on folded section

This will produce the folded shape shown in the lower half of figure A/10.

Is this fold pattern OK?

This shows that the folds from the two sides slightly overlap in the middle, and this might or might not be acceptable. Remember also that we have been working with an artificially large fold thickness so that we can see what we are doing: the folds look much better if we reduce this to a sensible value:

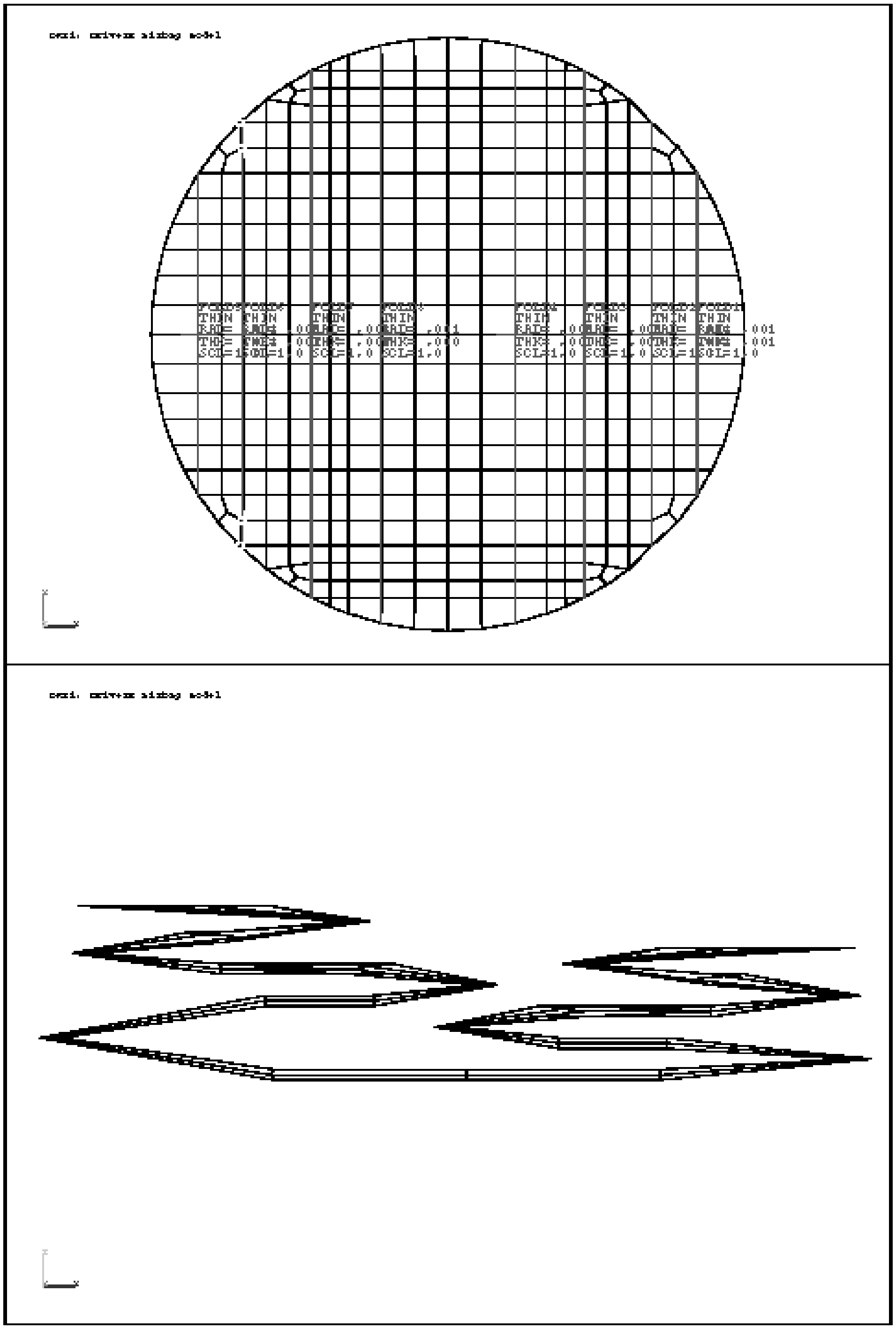
EXECUTE, ALL, 1.0e-3	To make min thickness 1.0mm
REFOLD	To see the effect of this

You can use ***MODIFY*** to change any attribute of any fold (see 5.1.6) if you want to change this pattern.

Executing the folds perpendicular to the Y axis

Refer back to figure A/4 and you will recall that there are 4 folds postulated perpendicular to the Y axis: Y1 to Y4.

These may be defined, checked and copied in exactly the same way as the X folds; and this is left as an exercise for the reader. Some hints:



Appendix II: Driver's bag. Effect of reflecting folds #1 to #4, and the resulting shape

figure **A/10**

- Start at the top with fold Y1, and make it a "down" fold;

- Swap between "up" and "down" between successive folds;

- Remember to "check" fold definitions. If you get some folds that look as if material is bulging out of the fold it is probably because you have failed to run through the "check" procedure and adjust nodal positions.

- The author found that his final (copied) fold Y8 (fold #16) just failed to pack down correctly at the 1mm thickness used. He used "modify" to adjust its **folded** position to a slightly more negative Y value of -0.11mm.

The author's final folded configuration is shown in figure A/11; and these are the fold definitions stored in the "saved" fold file:

```
/examples/driver_folds.sav
```

2.5.3 Reloading the pre-defined fold definitions

You can reload the saved, pre-defined fold definitions by using the commands:

[AIRBAGS], RELOAD, <filename>	To reload folds from disk

The <filename> you should give is:

/examples/driver_folds.sav

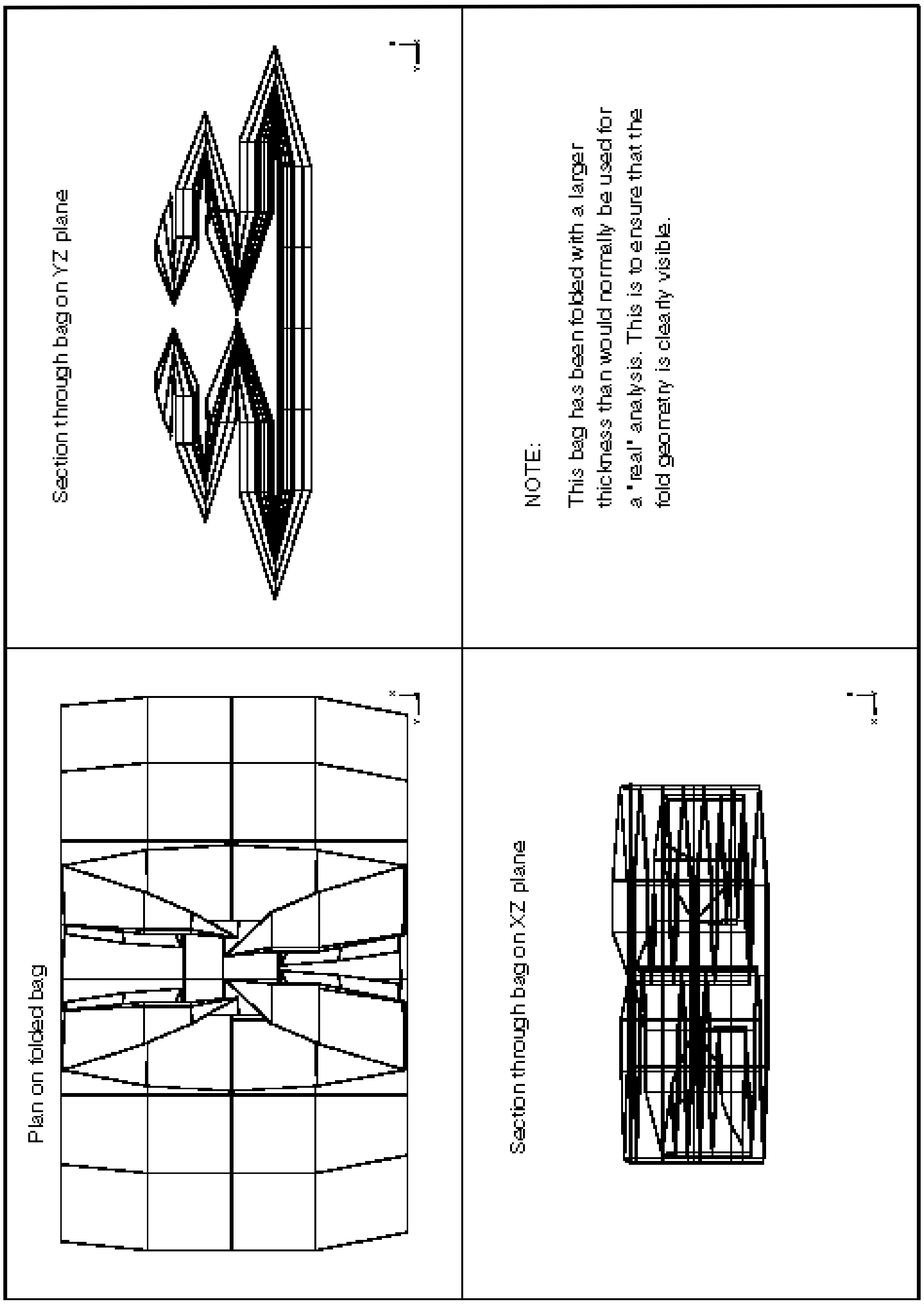
Which contains 16 fold definitions. This is in fact two files:

driver_folds.sav_bin	Binary files for reloading
driver_folds.sav_asc	Ascii fold definitions

The ascii fold definitions are formatted in the correct way for inclusion in an "ingrid" format input file, and will be useful for those generating airbags in that way. Being ascii it is a readable file and may be edited. It is never re-read back into INGRID so you can edit it with impunity.

The binary ones are used for reading back into INGRID. You will not be able to read or edit these.

Don't forget to ***CHECK*** your mesh if you import these definitions.



Section through bag on YZ plane

Plan on folded bag

Section through bag on XZ plane

NOTE:

This bag has been folded with a larger thickness than would normally be used for a "real" analysis. This is to ensure that the fold geometry is clearly visible.

Appendix II: The final driver's side bag in its folded configuration figure **A/11**

2.5.4 Changes required for a "real" bag

Note that this example has been folded with a thickness of 1mm so that the folds are visible. For a "real" analysis the author would tend to pack folds down to (say) 1/10th of this value, and then use a load-curve in the airbag (a13) contact to ramp up the effective contact thickness over (say) the first 5 mS.

2.6 Folding the passenger's side "pillow" bag

Refer to figure A/3 for the general geometry of this bag.

It is topologically a more complex structure:

- It is definitely not flat: the "mouth" presents a problem;
- It already has a "tuck" of material meshed into it.

So it is a bit harder to fold.

It is assumed that you are starting from the "dyna3d" format input deck:

```
/examples/pillow.inf
```

Enter the ***`AIRBAGS`*** menu in the data processing part of the code, and...

2.6.1 Defining what to fold

By default the whole model is selected for folding, but we can no longer tolerate that for this bag since the "mouth" cannot be folded. So we must remove it from the folding list. Do this by:

`SELECT, REMOVE, MAT, 3` To deselect material 3 (the "mouth")

Check that you have got this right:

`SKETCH` To see what is selected

You might care to experiment with what happens if you don't deselect the "mouth": you will find that it gets pulled up into the mesh when you apply the first fold.

2.6.2 Defining folds

As with the driver's side bag it is important to have a plan worked out beforehand: figure A/12 shows the plan for this mesh. And again it is better (if possible) to work from the outside of the mesh inwards.

In this example we will make the first fold the mesh line perpendicular to the X axis that has the most +ve X coordinate. This is fold X1 in figure A/12.

Define the first fold by:

```
DEFINE, NODE_NODE              To pick two nodes (see fig A/13)
       +                       To use +ve X side of line
        UP                     To fold it up over mesh
        1.00E-3                To use fold thickness 1mm
        1.0                    To use a scale factor of 1.0
        1.0                    To use a radius of 1mm
        THIN             To use a "thin" fold
```

At this point the fold will be sketched on the mesh as shown in figure A/13, and you will be asked if this fold definition is OK.

```
        OK                     Will confirm and store fold.
```

Check this fold by:

```
CHECK                          To check fold geometry.
```

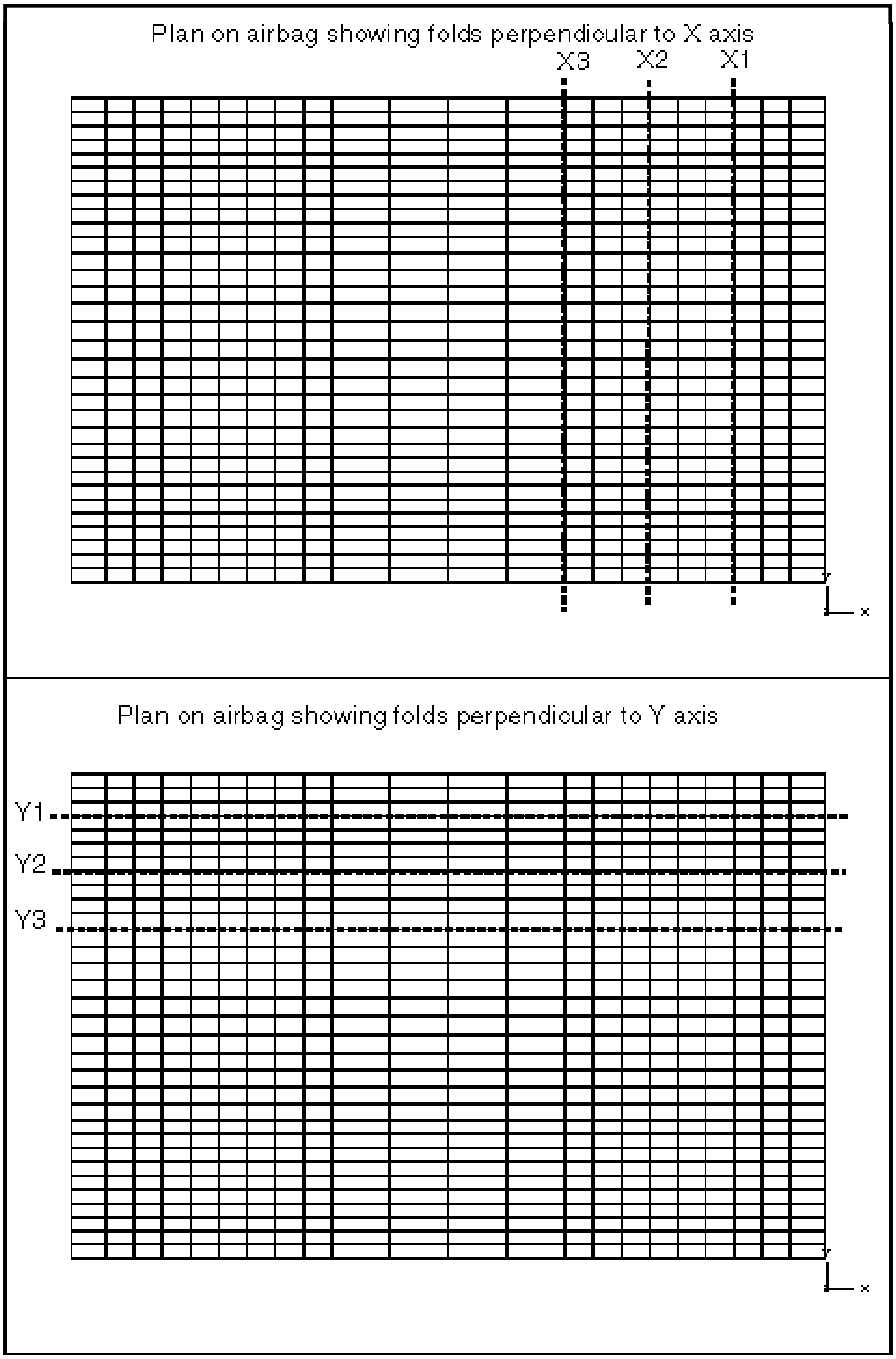
This should return the message **"Folds check out OK"**.

Execute this fold by:

```
EXECUTE, ALL             To make all folds active
         1.0e-3                To choose a min thickness of 1mm
```

We know that a thickness of 1mm will be hard to see, so re-execute fold with a thickness of 2.5mm to make it more visible:

```
EXECUTE, ALL             To make all folds active
         2.5e-3                To choose a min thickness of 2.5mm
```



Plan on airbag showing folds perpendicular to X axis

Plan on airbag showing folds perpendicular to Y axis

Appendix II: Folding plan for passenger side "pillow" airbag

figure **A/12**

Check the fold visually

Use any viewing command (eg ***POOR***) to draw the newly folded shape. If you choose a section on the XZ plane (command ***SXZ***) and zoom in you should see the image shown in the lower half of figure A/13.

Execute the remaining X folds

The same process can be carried out for folds X2 and X3. Figure A/14 shows the fold definitions and the resulting shape.

You should find no problems in defining these two folds: make X2 a "down" one, and X3 "up". ***CHECK*** them as before, but no errors should appear.

Adjust the fold scale factors

Because there is already a "tuck" fold meshed within the parent material, and the folding algorithms don't "know" about this, they tend to squeeze material together too tightly in the folds themselves. This is apparent if you zoom in onto any fold, and look at the spacing of the material.

Figure A/15 shows clearly what the default scale factor of 1.0 gives and how, for this mesh, increasing it to 4.0 gives a much better conditioned fold. Compare the two folds shown in the bottom right corner of this mesh: one using a scale factor of 4.0, and one with 1.0.

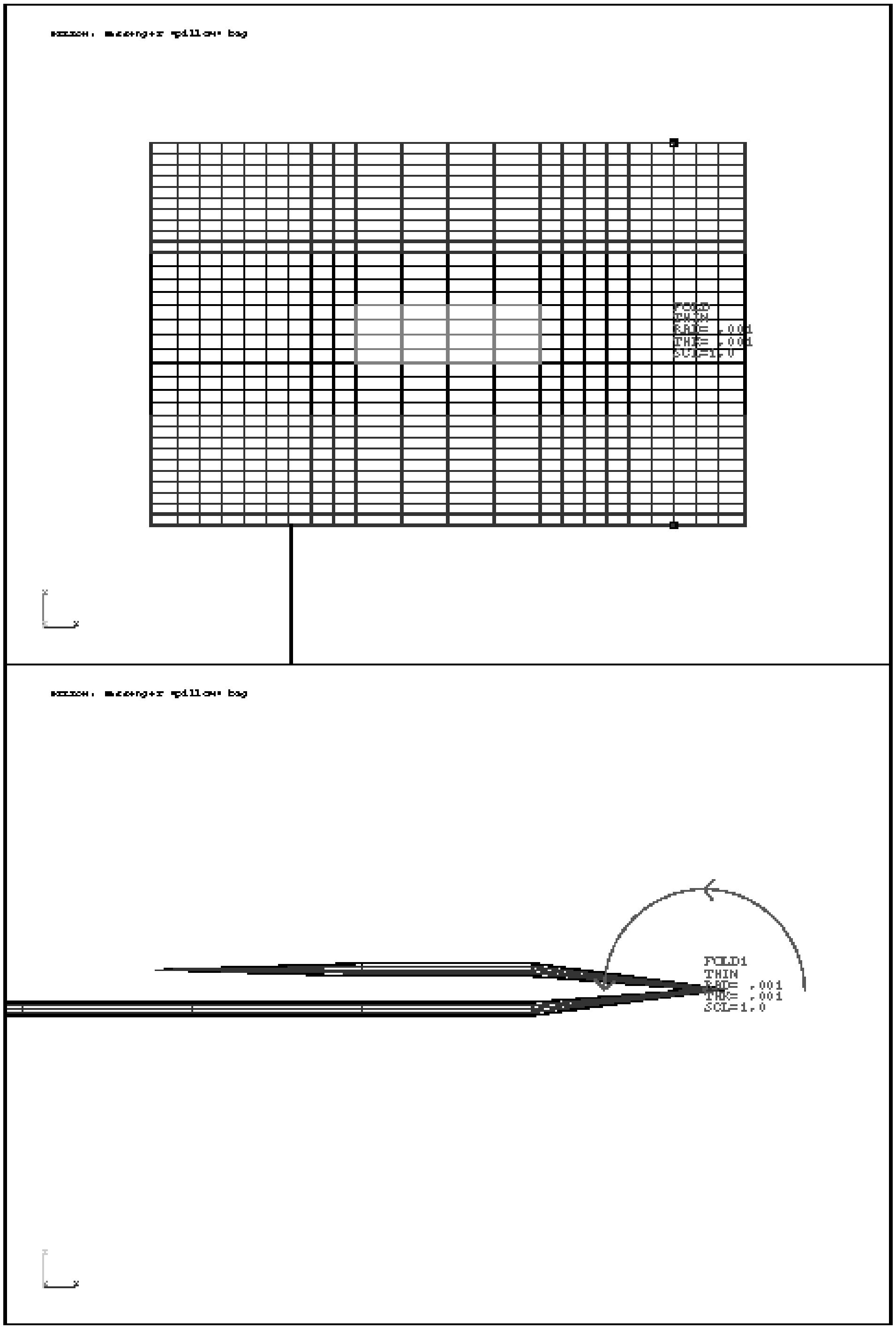
To adjust these factors, for each fold:

```
MODIFY, <fold no>              To select fold to modify
        V8,  4.0               To change scale factor to 4.0
        SAVE               To save changed fold definition
```

When using ***MODIFY*** try turning on the ***UPDATE*** button in the box at the top of the screen. This will cause all folds selected by ***EXECUTE*** to be redrawn every time you make a change. Changes due to the current fold are shown even though they have not been explicitly ***SAVE***d.

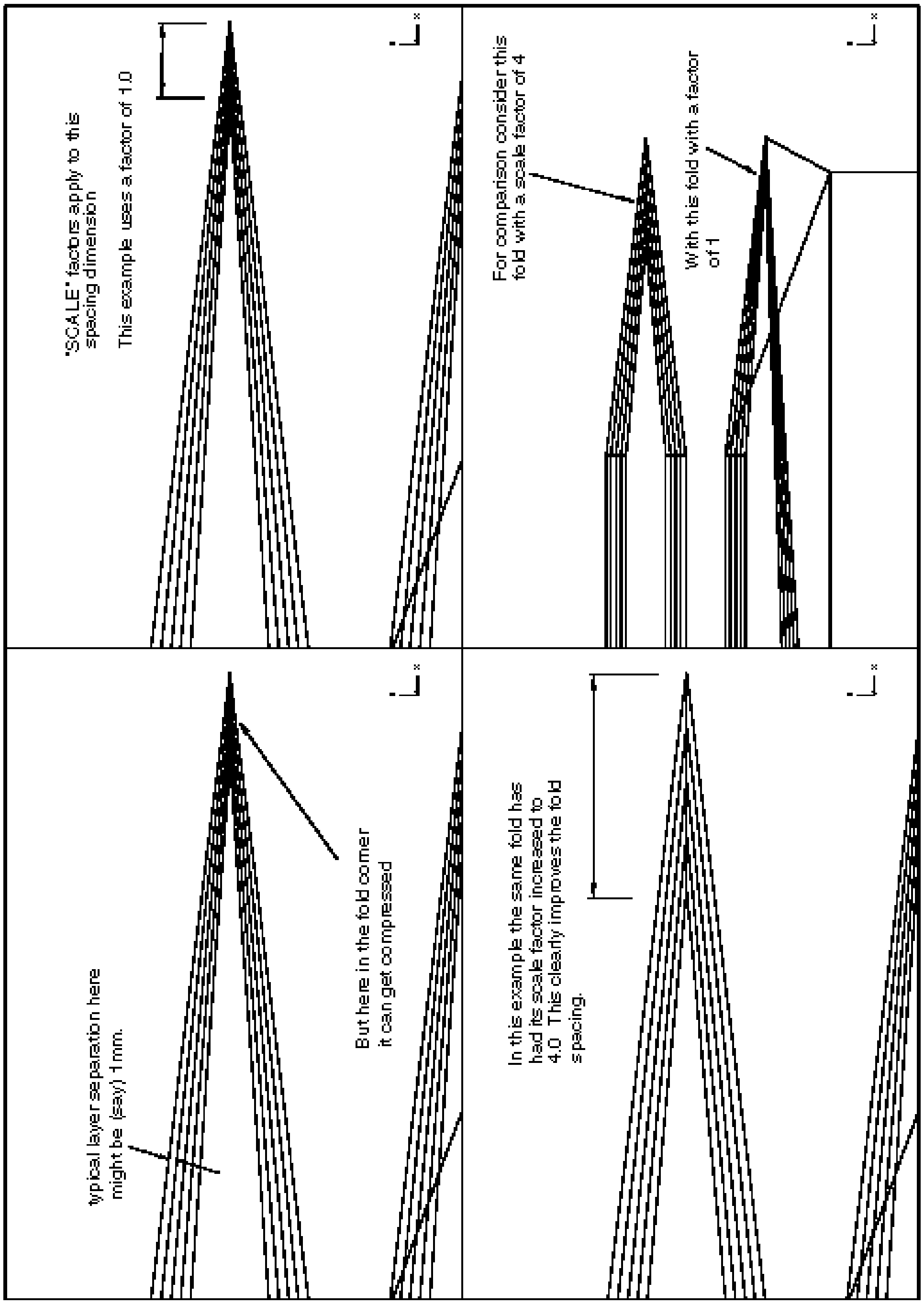
Turning this switch on is like issuing a ***REFOLD*** command every time you change something.

Remember that changes made with ***MODIFY*** only become permanent if explicitly ***SAVE***d, so you can experiment freely.



Appendix II: Passenger's "pillow" bag. Definition of first fold X1, and resulting shape.

figure **A/13**


'SCALE' factors apply to this spacing dimension
This example uses a factor of 1.0
For comparison consider this fold with a scale factor of 4
With this fold with a factor of 1
typical layer separation here might be (say) 1 mm.
But here in the fold corner it can get compressed
In this example the same fold has had its scale factor increased to 4.0 This clearly improves the fold spacing.

Appendix II: How material gets squeezed in corners, and using SCALE to fix this figure **A/15**

Copy X folds #1 to #3 to opposite side of bag

We need to copy folds #1 to #3 anti-symmetrically about the mesh centreline to form folds #4 to #6. This is simple:

```
COPY, ALL                     To copy all existing folds
      REFLECT                 To copy anti-symmetrically
      0.0                     0.0 is centre of bag
```

It is a good idea to check that the reflected folds produced this way are in the right place. This is most easily done on the unfolded geometry, so:

```
RESET                         To unfold bag
SXY; AC                       Autoscaled plan view on XY
SKFALL                  To sketch all folds
CHECK                         To check the new folds
```

Figure A/16 shows all X folds on the mesh, and section through the fold definitions.

Executing the folds perpendicular to the Y axis

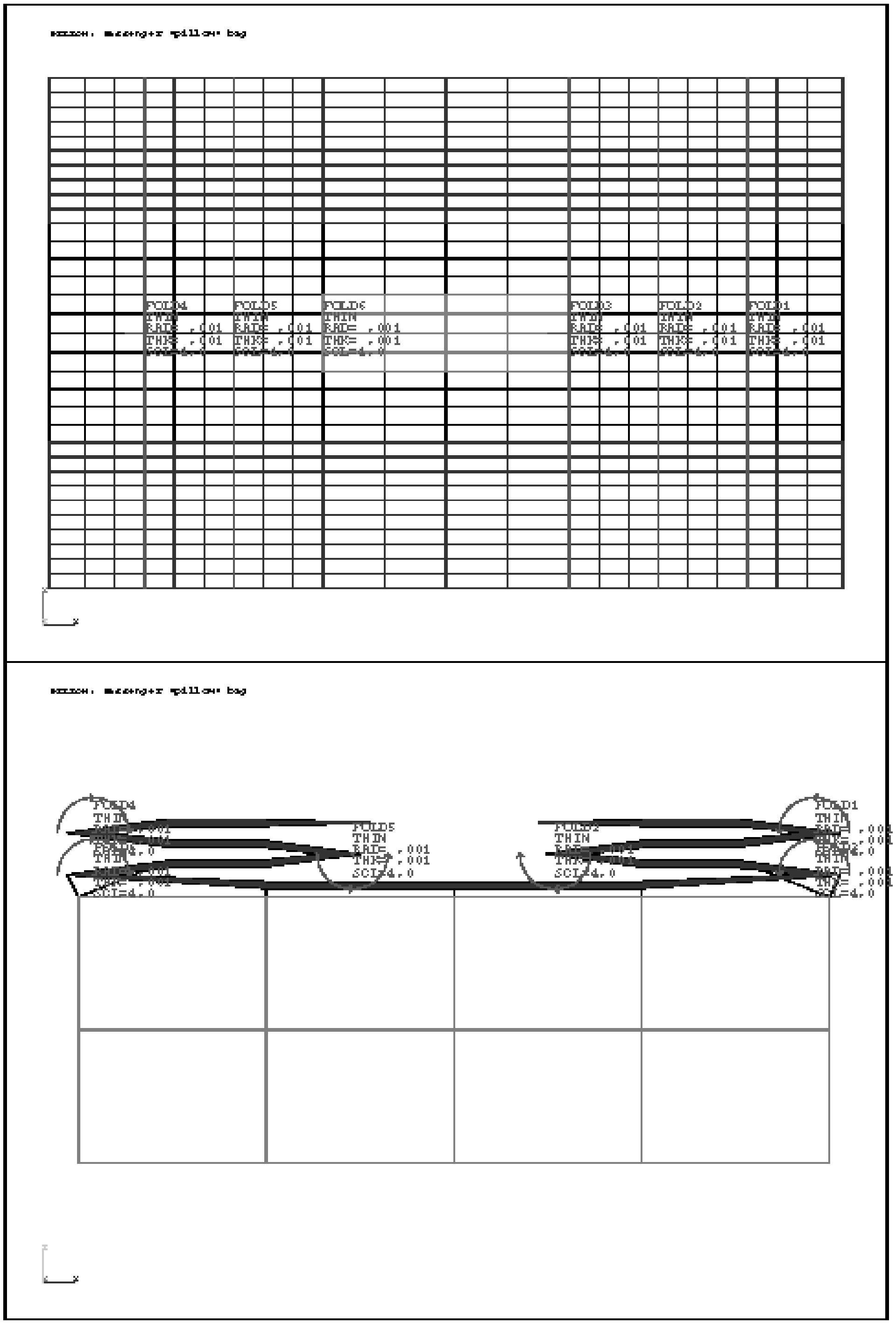
Refer back to figure A/12 to see that there are 3 folds postulated perpendicular to the Y axis: Y1 to Y3.

These may be defined, checked and copied in exactly the same way as the X folds, and figure A/17 shows what happens if you do this: clearly something has gone wrong!

What has happened is that the scale factor of 4.0 we used for the X folds has also been used for Y, and it has overdone the spacing of these folds. So we must change it back:

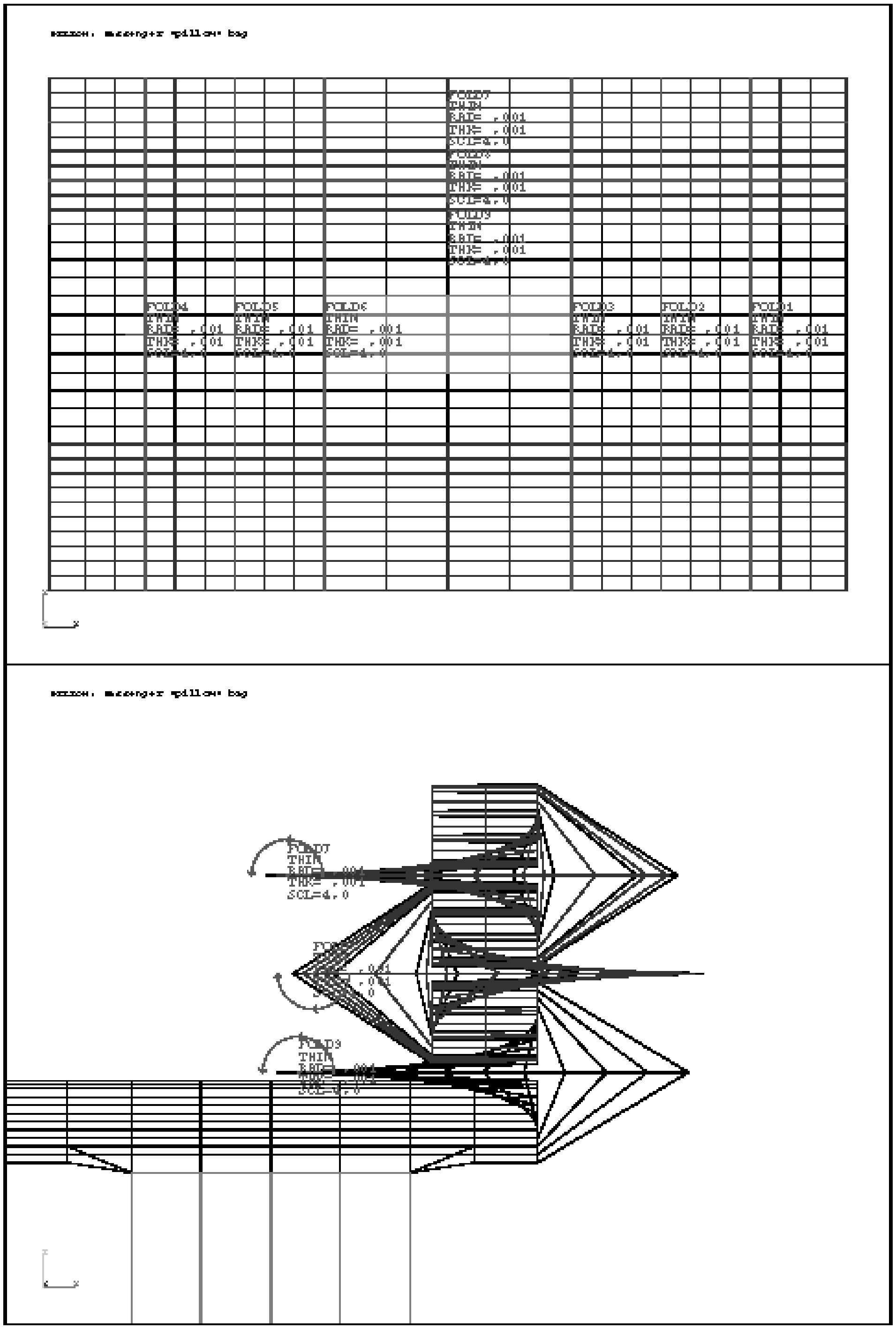
```
MODIFY, FOLD_7                This is the fold Y fold (Y1)
        V8, 1.0               Select scale factor 1.0
        V8, 0.5               No good, try 0.5
        SAVE            Save this
```

The author found that a scale factor of 0.5 was suitable for folds #7 and #8 (ie Y1 and Y2), but that 0.6 was better for fold #9 (Y3). By using the ***UPDATE*** switch it was possible to try a range of different values and to see exactly what the best one was. See figure A/18 for the final shape.

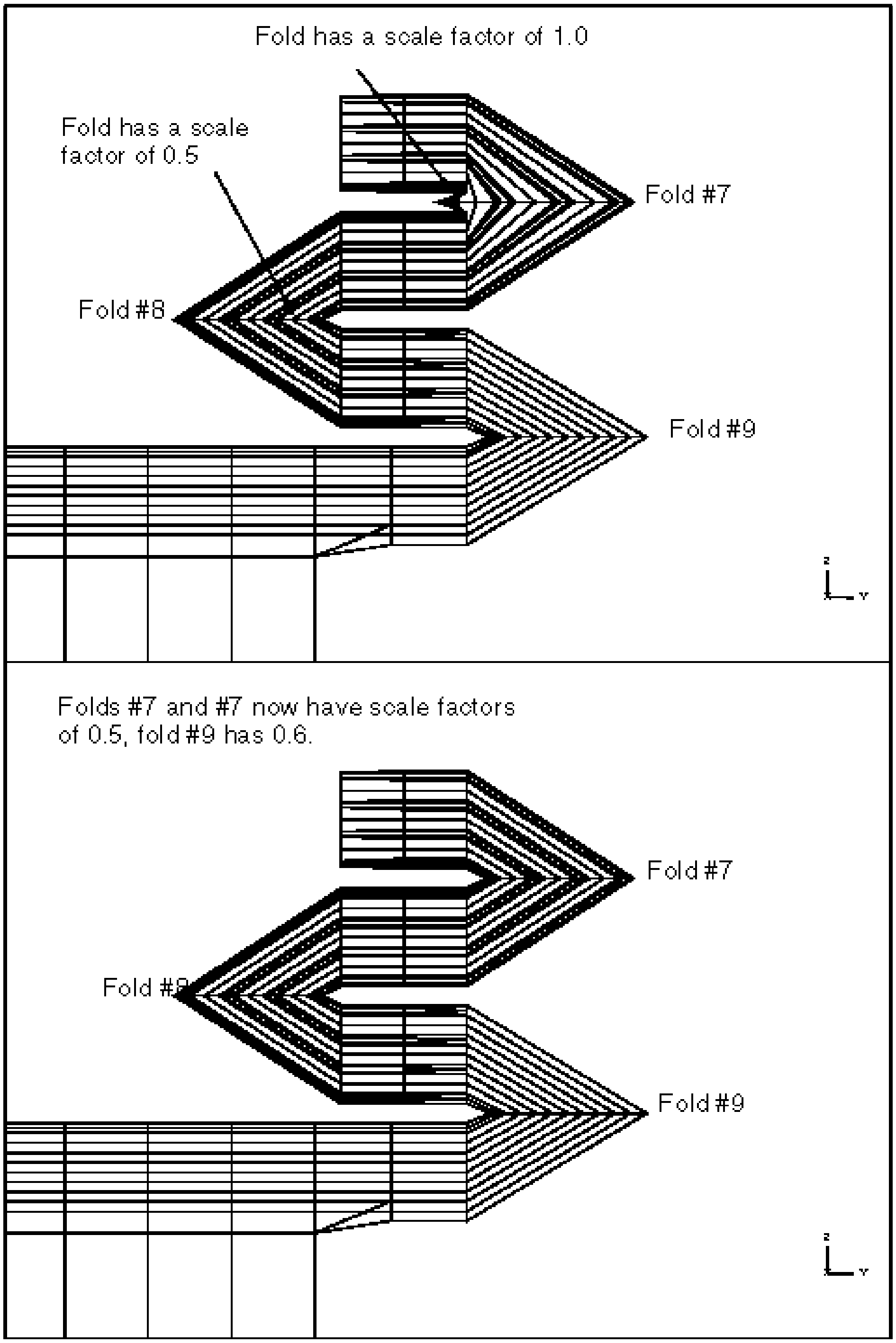


Appendix II: Passenger pillow bag: all X folds defined and the resulting folds. (nb: thickness = 2.5mm)

figure **A/16**



Appendix II: Passenger "pillow" bag: initial definition of folds Y1 to Y3 – Whoops! What has gone wrong? figure **A/17**


Fold has a scale factor of 1.0
Fold has a scale factor of 0.5
Fold #7
Fold #8
Fold #9
Z
Y
Folds #7 and #7 now have scale factors of 0.5, fold #9 has 0.6.
Fold #7
Fold #8
Fold #9
Z
Y

Appendix II: Passenger "pillow" bag, sorting out Y fold scale values

figure **A/18**

Copy basic Y folds to opposite side

As with the X folds the Y ones (Y1 to Y3) must now be copied anti-symmetrically across the centreline. This is carried out as explained previously.

And reset the thickness value to something sensible

We have been working with a global minimum thickness of 2.5mm, but for a passenger bag a value of (say) 0.5mm would be more representative. In fact we have defined all folds to have the default thickness of 1mm, so this will be taken as the minimum unless you ***MODIFY*** each fold. But anyway:

```
EXECUTE, ALL            For all folds
         0.5e-3               Use 0.5mm
         REFOLD               To see the effect
```

And we have a more plausible bag as shown in figure A/19.

Is this acceptable?

It is clear from figure A/19 (top) that the planned Y folds could probably be packed in more towards the centre of the bag. So how do you move a fold location?

Again, use ***MODIFY***, but this time update the fold coordinates. First get a plan view of the unfolded mesh with the folds sketched on it, so you can see what you are dealing with:

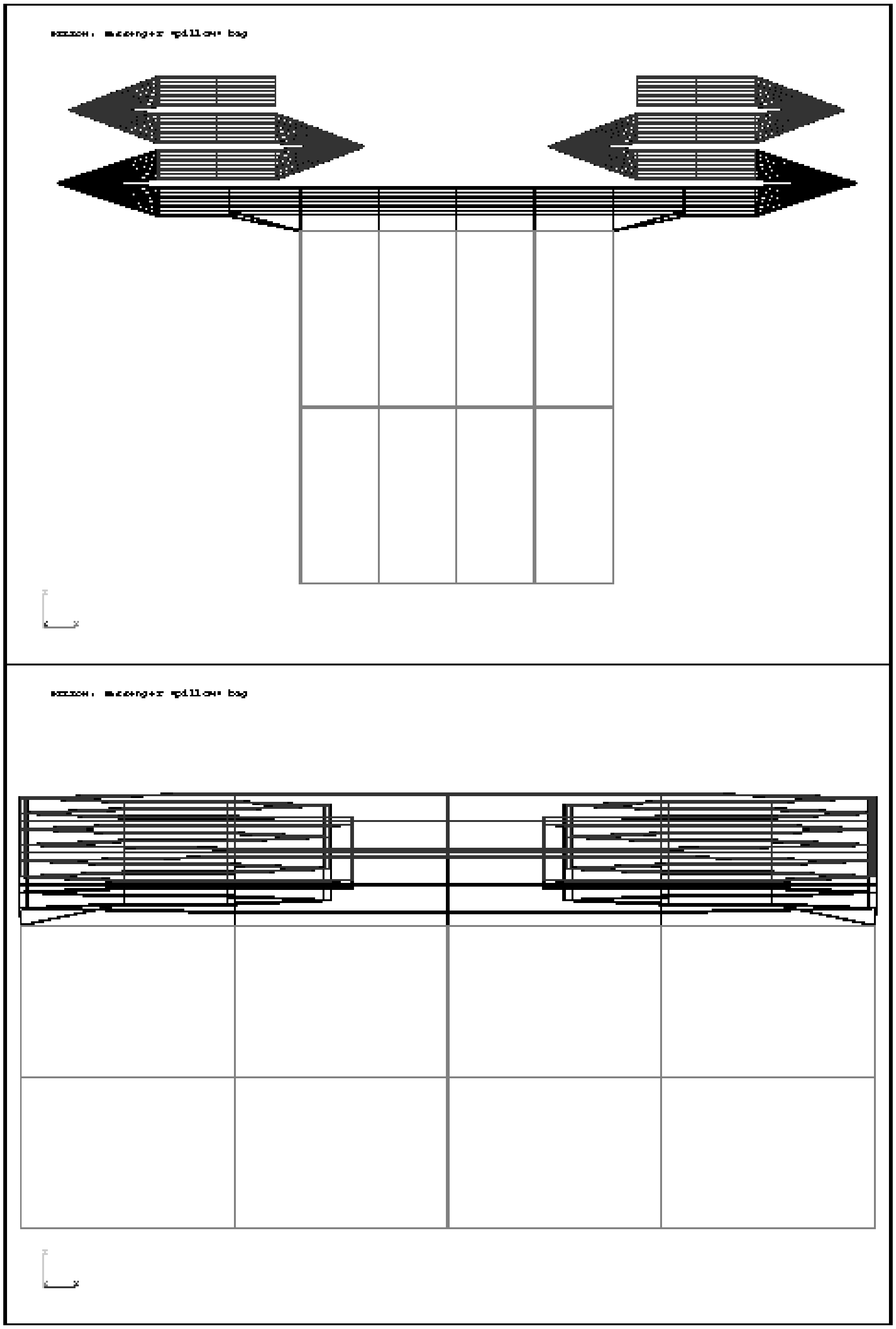
```
RESET                         Undo all folds
SXY; AC                       Get an autoscaled plan on XY plane
SKFALL                  Sketch all folds
```

Then modify fold #7 (fold Y1) as follows:

```
MODIFY, FOLD_7                Select first Y fold (Y1)
        V1, NODE              Choose unfolded position by node pick
```

And pick a node on the next mesh line in towards the bag centre. Then you must change the "folded" position as well to be the same:

```
        V2, AS_UN             Make "folded" the same as "unfolded"
        SKETCH                To sketch new position on mesh
        SAVE            Save changes
```


[illegible] passenger "pillow" bag
[illegible] passenger "pillow" bag

Appendix II: Passenger "pillow" bag: First pass at final folded shape. Elevations on YZ and XZ planes. figure **A/19**

Repeat this for folds #8 and #9: move each one mesh line further in towards the bag centre.

Note:

- If you left the ***UPDATE*** switch on in ***MODIFY*** you probably cursed the first time you updated a coordinate: the whole mesh would be refolded and you wuldn't be able to see what you were doing. This is why it is a switch and not a default!

- Exit modify (use ***QUIT*** to abandon changes), go through the ***RESET, SXY, AC, <draw mode>, SKFALL*** procedure again to redraw the mesh. Re-enter ***MODIFY*** again, and **this time** turn the ***UPDATE*** switch off before changing anything.

Figure A/20 shows a plan on the unfolded mesh plus the folded shape with folds #7, #8 and #9 adjusted inwards like this, but #10 to #12 left as they were. This is to let you compare the difference in the fold patterns on either side of the centreline.

<u>The final configuration</u>

Figure A/21 shows the author's final mesh with all Y folds brought more in towards the centre.

These fold definitions stored in the "saved" fold file:

```
/examples/pillow_folds.sav
```

2.6.3 Reloading the pre-defined fold definitions

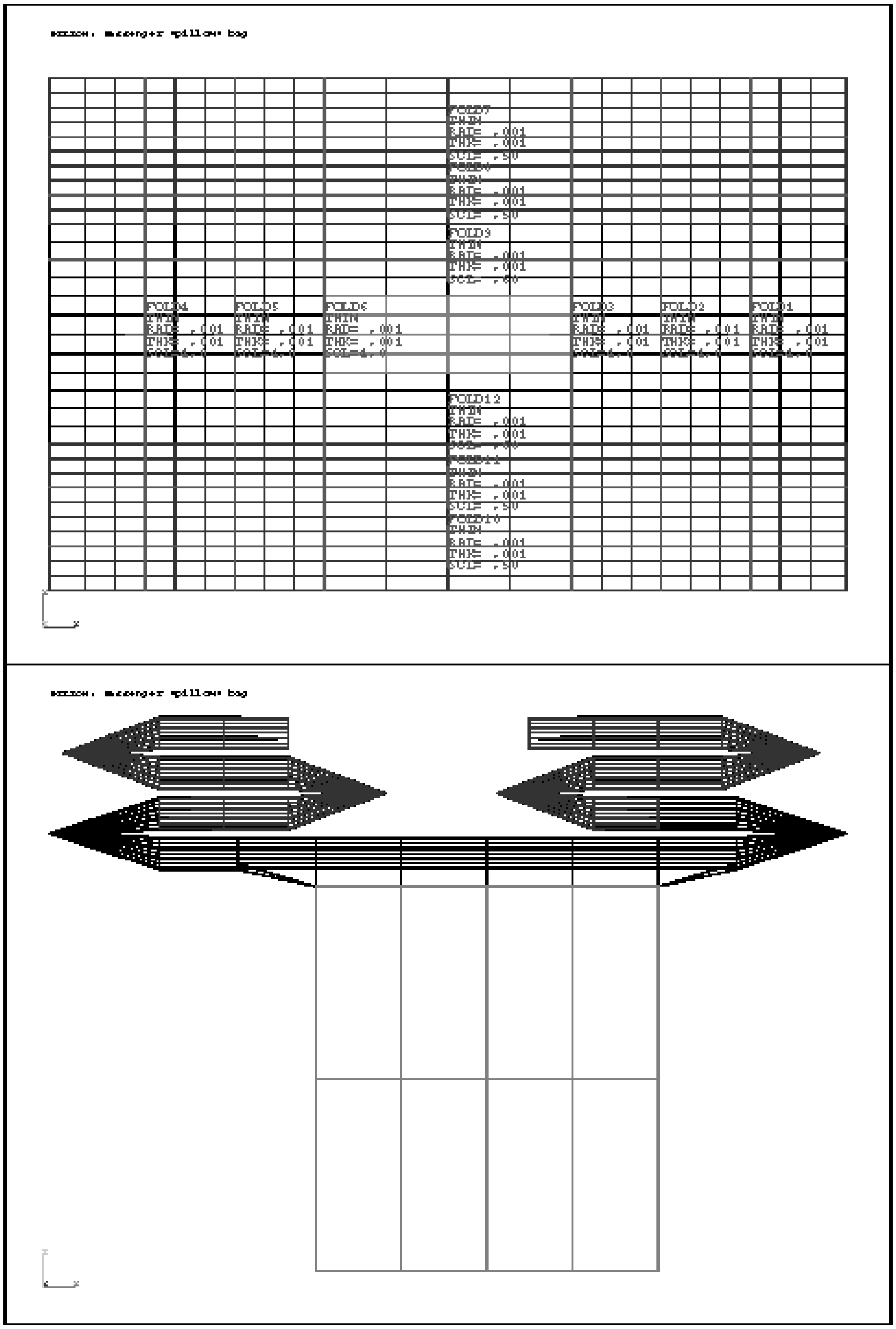
You can reload the saved, pre-defined fold definitions by using the commands:

[AIRBAGS], RELOAD, <filename> To reload folds from disk

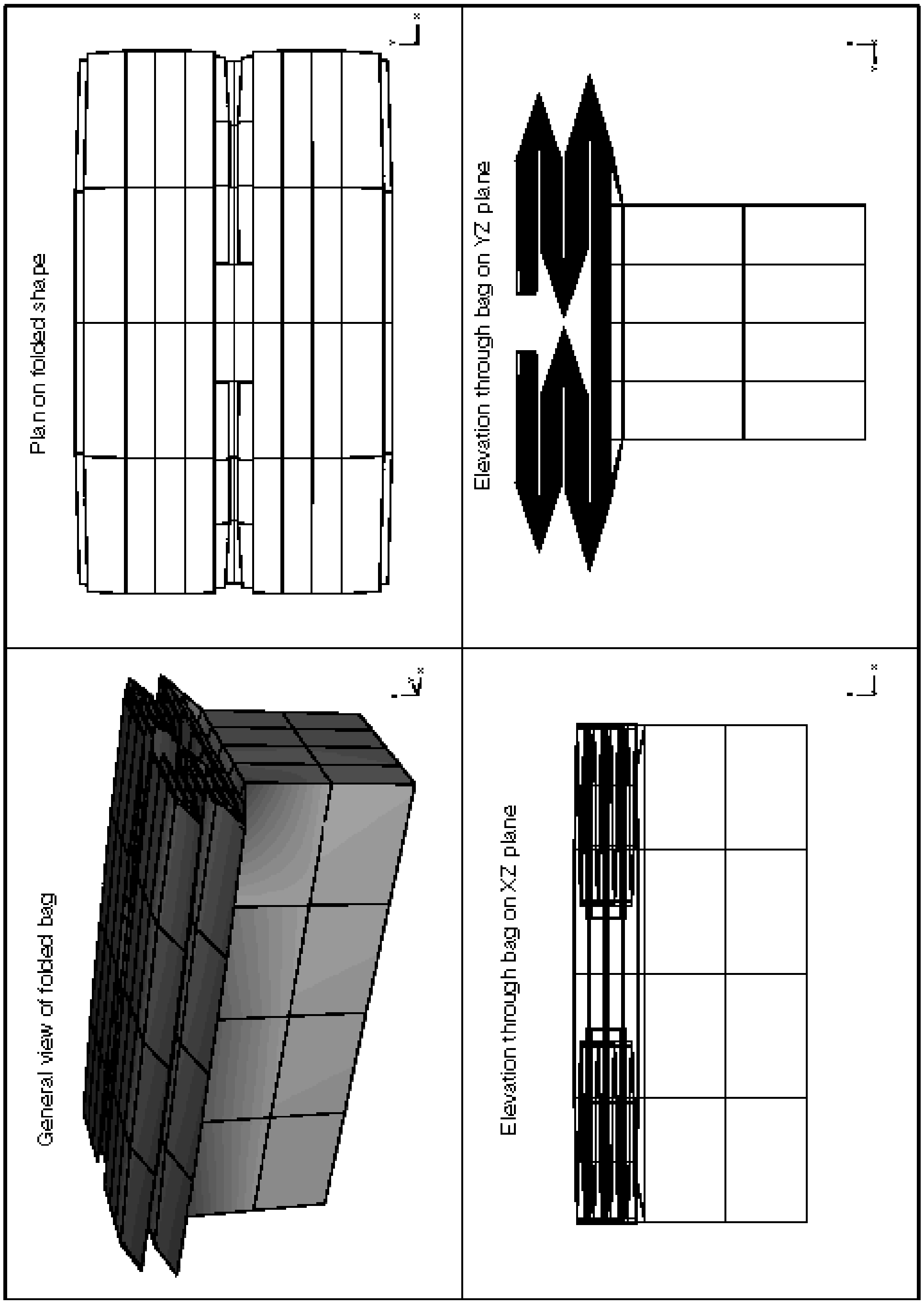
The <filename> you should give is:

```
/examples/pillow_folds.sav
```

Which contains 12 fold definitions. See the notes on the driver bag (2.5.3) on what these files are and how to use them. Don't forget to remove the "mouth" material from the folding list, and to ***CHECK*** your mesh!



Appendix II: Passenger "pillow" bag. First three Y folds revised, second three left alone. Compare shapes figure **A/20**



General view of folded bag

Plan on folded shape

Elevation through bag on XZ plane

Elevation through bag on YZ plane

Appendix II: Passenger "pillow" bag. Final shape figure **A/21**

2.6.4 Changes required for a "real" bag

The fold pattern is perhaps not optimal in Y: possibly it would be better to have four smaller folds, but care would be needed to see that they didn't meet in the middle.

The thickness of 1mm used for all folds here is probably too much for a passenger bag material: 0.3mm to 0.5mm would be more typical. You might like to go through the folds modifying them to see what the effect would be.

3.0 Do these analyses work?

Not an unreasonable question at all. The answer is yes, although they could use some tuning since they use notional parameters.

To use them in "real" jobs you will need to provide:

- Correct material definitions;

- Correct control volume definitions.

The values provided in the example "dyna3d" input files will run, but they are indicative only. See 3.1 and 3.2 below.

3.1 Results from a complete driver's side analysis

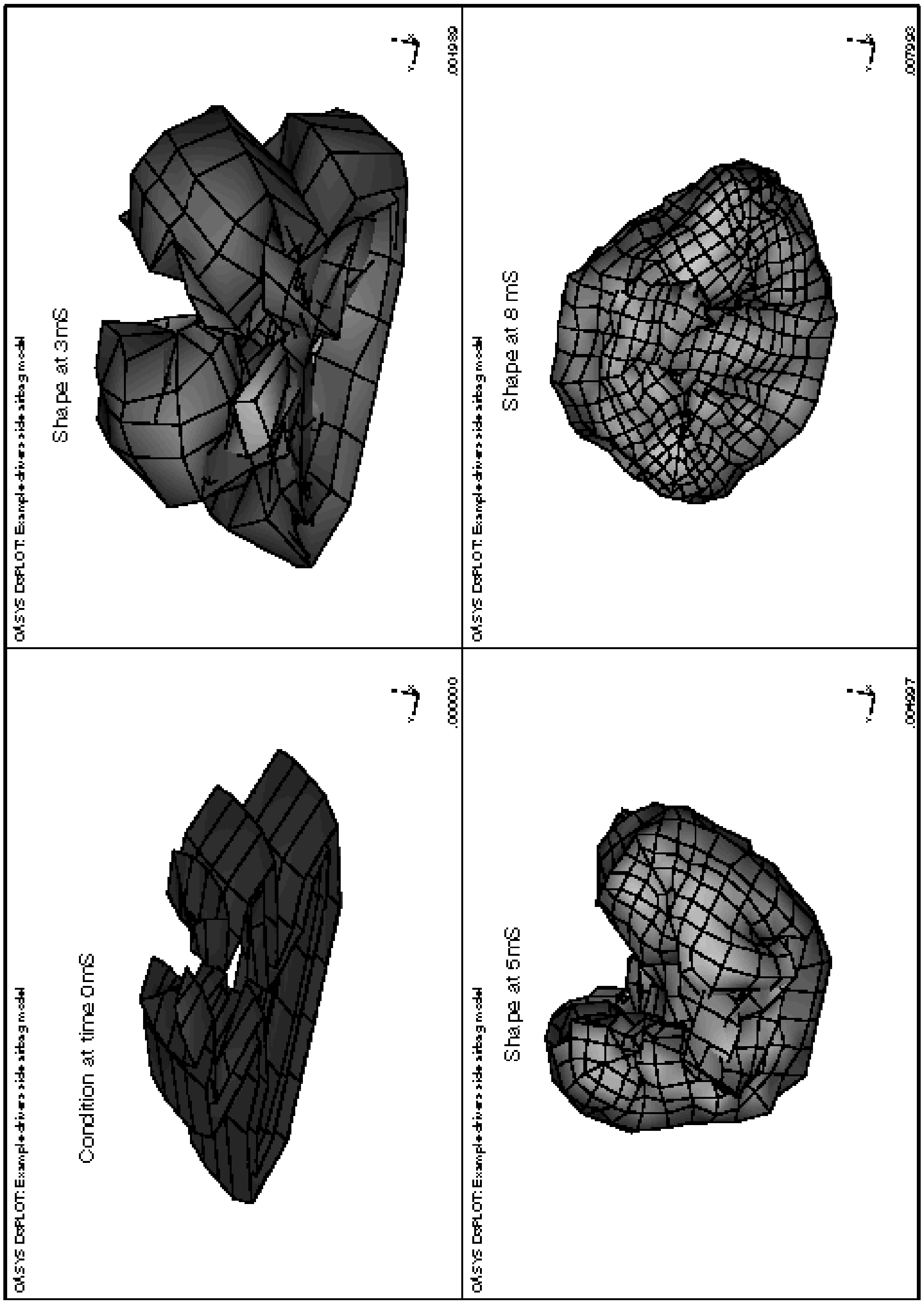
A complete, folded driver's side input deck using the folds defined above is included with INGRID. This is file:

```
/examples/driv1.inf
```

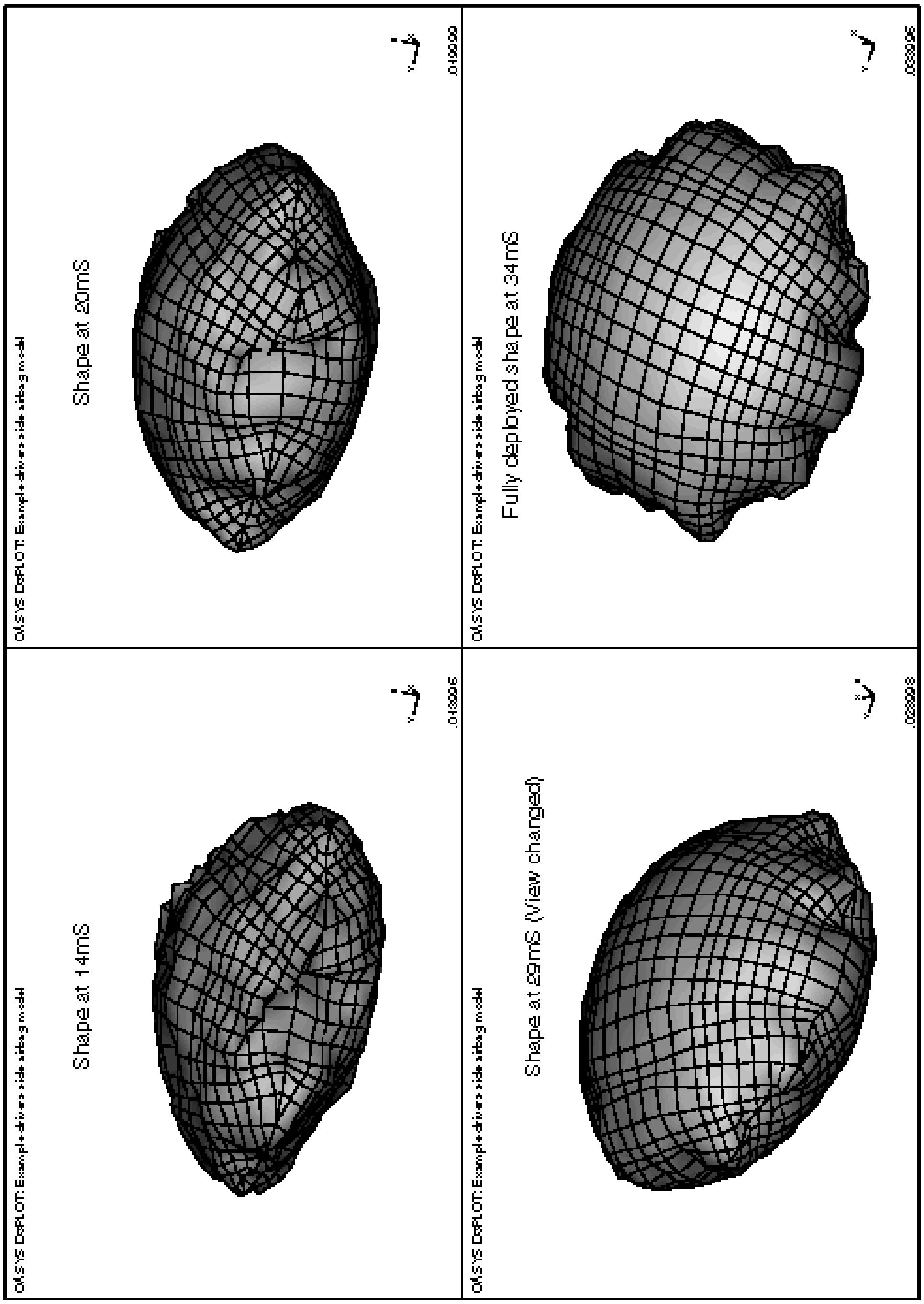
This is what you should get if you apply the pre-defined folds to the basic driver's side bag (**`/examples/driver.inf`**) used above.

The results of inflating it in isolation are shown in figures A/22 and A/23. These show that the control volume has insufficient venting.

This problem took 6 minutes to run to 50mS problem time on an HP715/100 in LS-DYNA3D 930.


OASYS D3PLOT: Example drivers side airbag model
Condition at time 0mS
.000000
OASYS D3PLOT: Example drivers side airbag model
Shape at 3mS
.001989
OASYS D3PLOT: Example drivers side airbag model
Shape at 5mS
.004997
OASYS D3PLOT: Example drivers side airbag model
Shape at 8 mS
.007993

Driver's side bag: Examples of actual deployment **figure A/22**


OASYS D3PLOT: Example drivers side airbag model
Shape at 14mS
.013996
OASYS D3PLOT: Example drivers side airbag model
Shape at 20mS
.019998
OASYS D3PLOT: Example drivers side airbag model
Shape at 29mS (View changed)
.028998
OASYS D3PLOT: Example drivers side airbag model
Fully deployed shape at 34mS
.033996

Drivers side bag: results of deployment

figure **A/23**

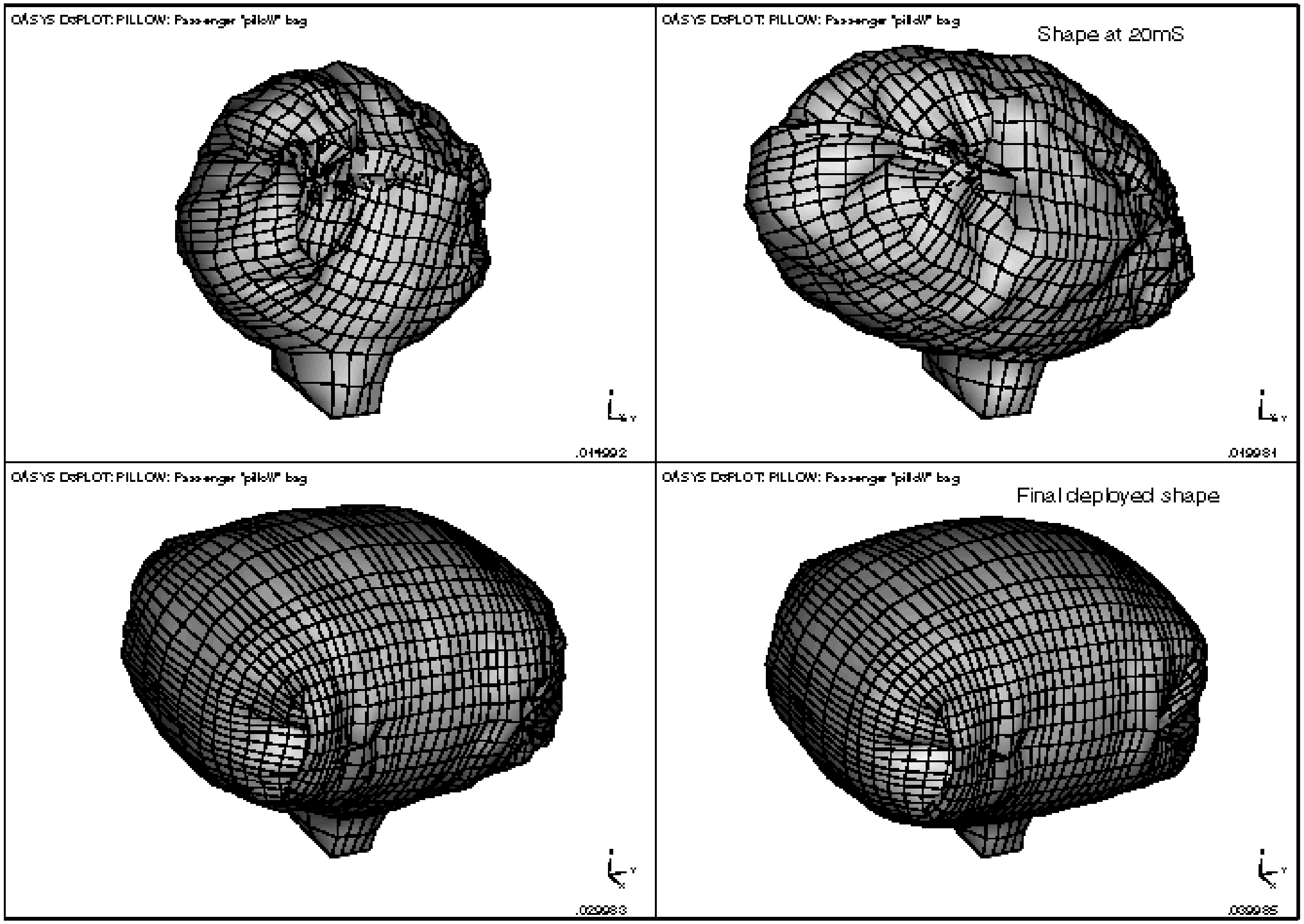
3.2 Results from a complete passenger side analsysis

You will also find a complete input deck for a passenger side bag in file:

```
/examples/pass1.inf
```

This is what you should get if you apply the pre-defined folds to the basic driver's side bag (**`/examples/pillow.inf`**) used above.

The results of inflating it in isolation are shown in figures A/24 and A/25. These also show that the control volume may have insufficient venting, although it is hard to tell when no occupant is available to hit it.


OASYS D3PLOT: PILLOW: Passenger "pillow" bag
.014992
OASYS D3PLOT: PILLOW: Passenger "pillow" bag
Shape at 20mS
.019981
OASYS D3PLOT: PILLOW: Passenger "pillow" bag
.029983
OASYS D3PLOT: PILLOW: Passenger "pillow" bag
Final deployed shape
.039985
X
Y

Passenger bag: example of actual deployment

figure **A/25**

APPENDIX III

Worked Examples of Seat-belt Positioning

APPENDIX III Worked examples of Seat-belt Positioning

1.0 Description

Generally you will start from a standard occupant model and a vehicle model. We will asume this here. The occupant used will be the Oasys RIGID HYBRID III dummy (RH3) referred to in Appendix I, and the car model is an extremely crude one (using a real car model would not be acceptable in a published document.)

2.0 Reading in the data

Read in the vehicle model in the normal way: see section 4.2.1 of this manual. In this example the standard file:

`/examples/chas.inf` (Which is in "oasys" format)

Has been used as the vehicle.

Read in the occupant model: see section 4.2.2 of this manual. In this case the ***`RH3`*** dummy model is used. Figure A/26 shows the initial orientation of dummy and car.

3.0 Position the occupant correctly in the vehicle

Enter the **`OCCUPANT`** section of INGRID, and use the options as described below.

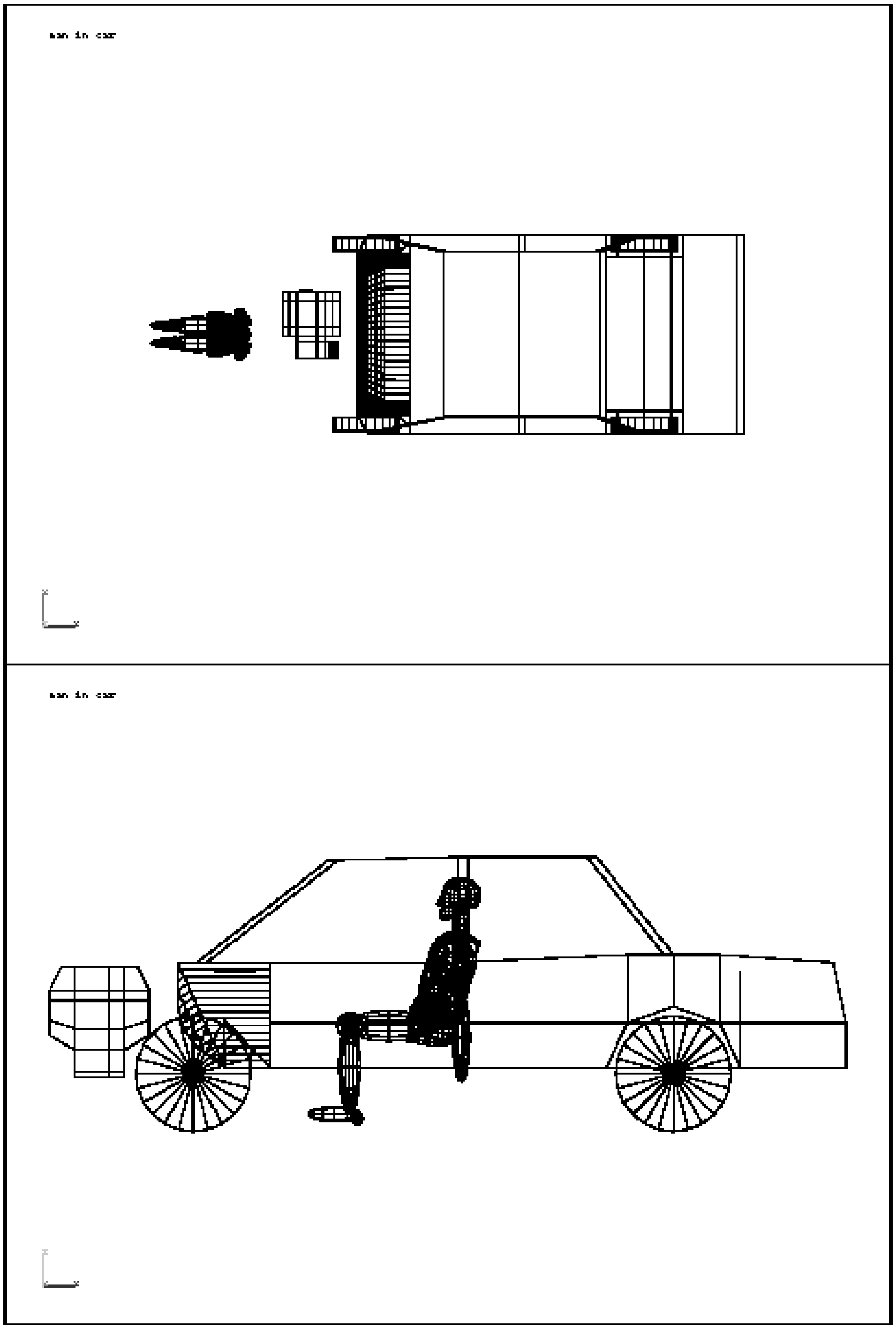
3.1 Set the occupant units correctly

It is possible that your vehicle and the standard dummy model are in different units systems. If this is the case modify the dummy using:

`[OCCUPANT], SCALE, <option>`	<option> being a units conversion factor.

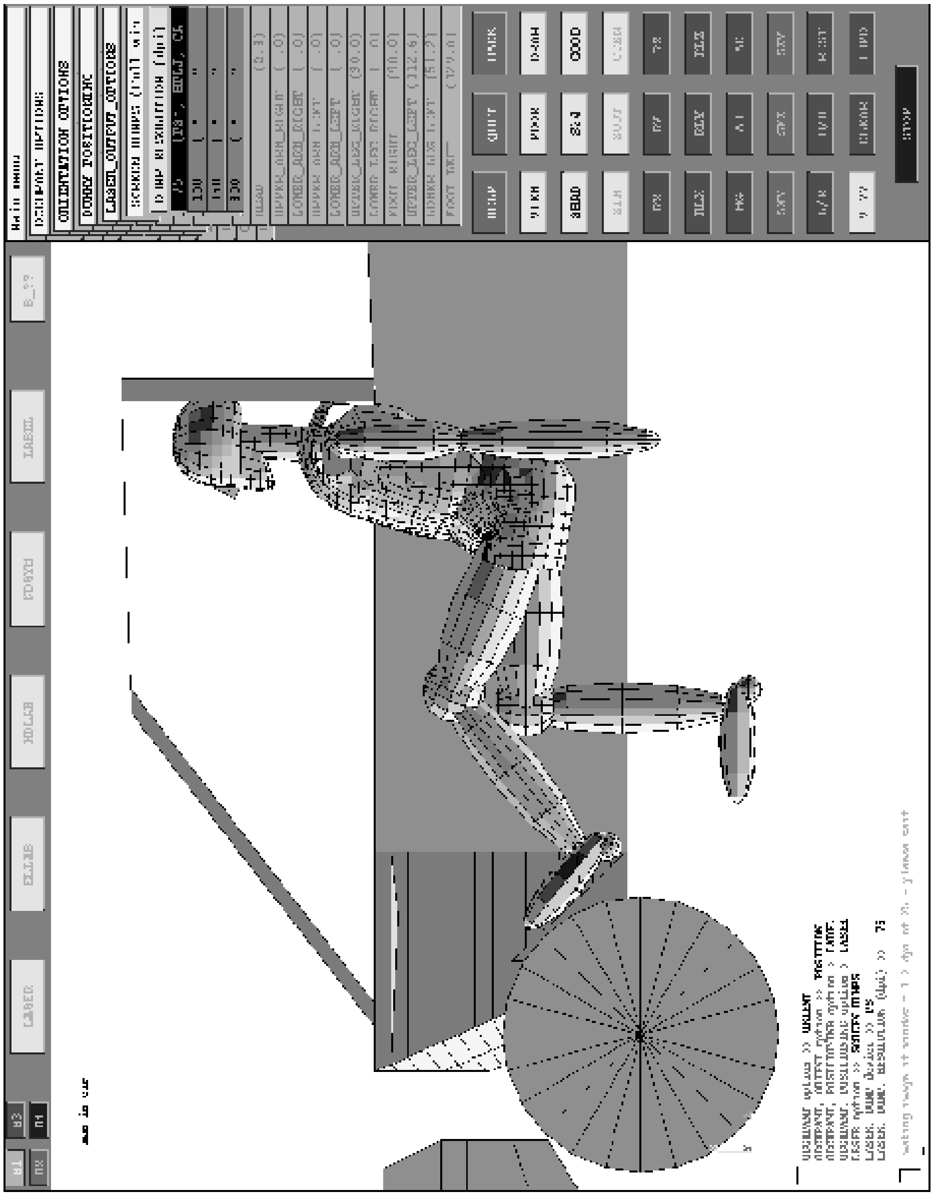
Typically you will be converting between "modified metric" and "SI". This procedure is defined in more detail in section 5.2.3 of this manual.

You will not need to do this for the models used here, they are both defined in "modified metric" units: mm, Te, s.



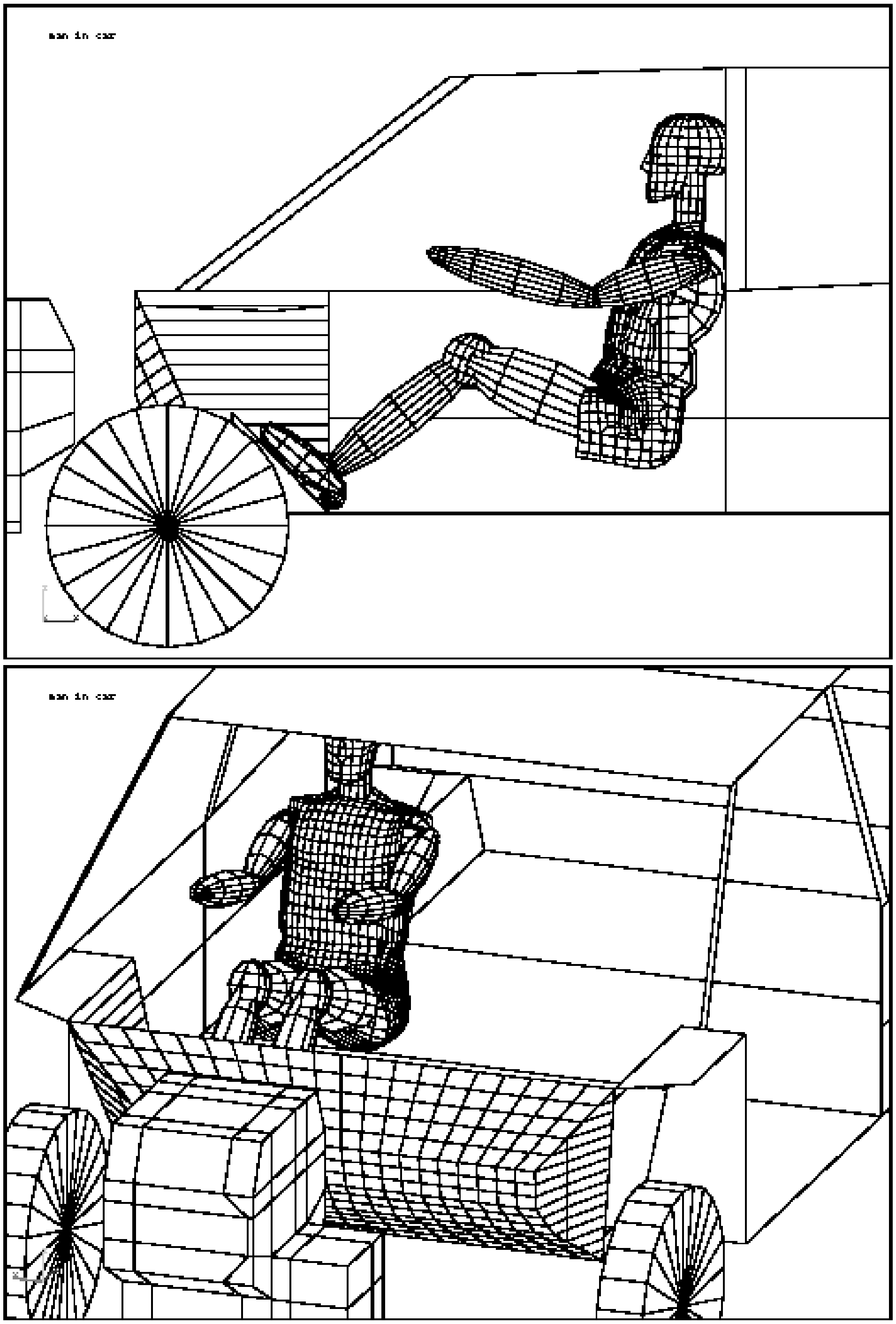
Appendix III: Seat-belt positioning. First stage: position the dummy in the vehicle

figure A/26



Appendix III: Seat-belt tensioning: Putting the dummy into position using the "drag and drop" method

figure **A/27**



Appendix III: Seat-belt positioning: Dummy positioned
in vehicle with all limbs arranged correctly

figure **A/28**

3.2 Move the occupant H-Point to the correct position

This is done using the ***ORIENT*** option:

ORIENT, TRANSLATE, <option>	To translate by [dx,dy,dz]

Where ***<option>*** may be:

XYZ_USER	If you know the offsets;
NEW_H_POINT	If you know the new H-Point coordinates;
N<option>	To move by a vector component between nodes.

Here we do not know the H-point, so we will move the dummy by translating it in:

NXY->NXY To move it in the XY plane. The author (who is English and so drives on the left) picked a node on the dummy's right shoulder, and moved it to a node on the car's "B" pillar.

NZ->NZ Having then changed the view to an elevation on XZ (using ***SXZ***). The author moved a node on the dummy's hip to a node on the cars waist rail.

This moved the dummy to the position shown on the lower half of figure A/26.

It then needed a bit of fiddling about to get it 100mm further forward (-X), and 50mm nearer the centre of the car (-Y):

TRANSLATE, -100 -50 0	To move it by this amount.

This located its H-Point at [2427, 460, 320] mm.

3.3 Position the dummy limbs in the correct attitude

First get a good view of the dummy in the vehicle:

- The author first got an elevation view on the XZ plane (***SXZ***);

- He then ***BLANK***ed those parts of the side of the car that were getting in the way.

Then he proceeded to ***DRAG*** limbs into a suitable position using:

```
[OCCUPANT], ORIENT, POSITION, DRAG          For arms and legs
```

This has to be done by eye. Refer to section 5.2.4 of this manual for help, and see figure A/27 which shows a typical screen layout during dummy limb positioning.

The X axis angles used by the author were:

LOWER_TORSO	**22.3 deg**	}
SPINE	**22.3 deg**	}
UPPER_TORSO	**22.3 deg**	} Unchanged from defaults
NECK	**2.3 deg**	}
HEAD	**2.3 deg**	}
UPPER_ARM_RIGHT	**72.9 deg**	
LOWER_ARM_RIGHT	**105.9 deg**	
UPPER_ARM_LEFT	**72.9 deg**	
LOWER_ARM_LEFT	**105.9 deg**	
UPPER_LEG_RIGHT	**112.6 deg**	
LOWER_LEG_RIGHT	**51.2 deg**	
FOOR_RIGHT	**129.6 deg**	
UPPER_LEG_LEFT	**112.6 deg**	
LOWER_LEG_LEFT	**51.2 deg**	
FOOR_LEFT	**129.6 deg**	

Y and Z angles were all zero. Figure A/28 shows the final position of the dummy in the car with all limbs at the correct angles.

3.4 Define a seat-belt

Enter the ***BELTS*** section and define a seat-belt line as follows:

<u>Define a seat-belt material</u>

You should strictly have a load-curve for loading, and another for unloading before defining a belt material. The author cheated and used two curves at random. These could be fixed later:

```
[OCCUPANT], BELTS, MATL_BELTS, CREATE       To create a belt material
                   <load curve #1>          For loading
                   <load curve #2>          For unloading
                   0.100e-6                 Mass / unit length
                  5.0                       Minimum length
                   OK                       OK to save material #1
```

Define a crude belt line

Once a belt material exists we can create a line. Do this by:

```
[OCCUPANT], BELTS, SEAT_BELTS, CREATE      To create a belt itself
                   USER_XYZ                To define 1st point
                   2680 815 925            Coord of first point
```

This coordinate is half way up the B pillar. There is no node there, so its location was obtained by using the **MEASURE, COORD** option in rotating button box B2 at the top of the screen. The coordinates of the nodes at top and mid-point of that pillar were averaged.

```
NODE->NODE        Mode is now node-node
<pick node>
<pick node>
<pick node>
<pick node>
      DONE        To finish picking nodes
2600 250 300            "D" ring position
```

Subsequent points across the chest and torso were selected by picking a line of nodes, although the "D" ring position was again estimated by measuring coordinates to be [2600, 250, 300]. Figure 29 shows the path across the chest and the location of the "D" ring behind the left pelvis.

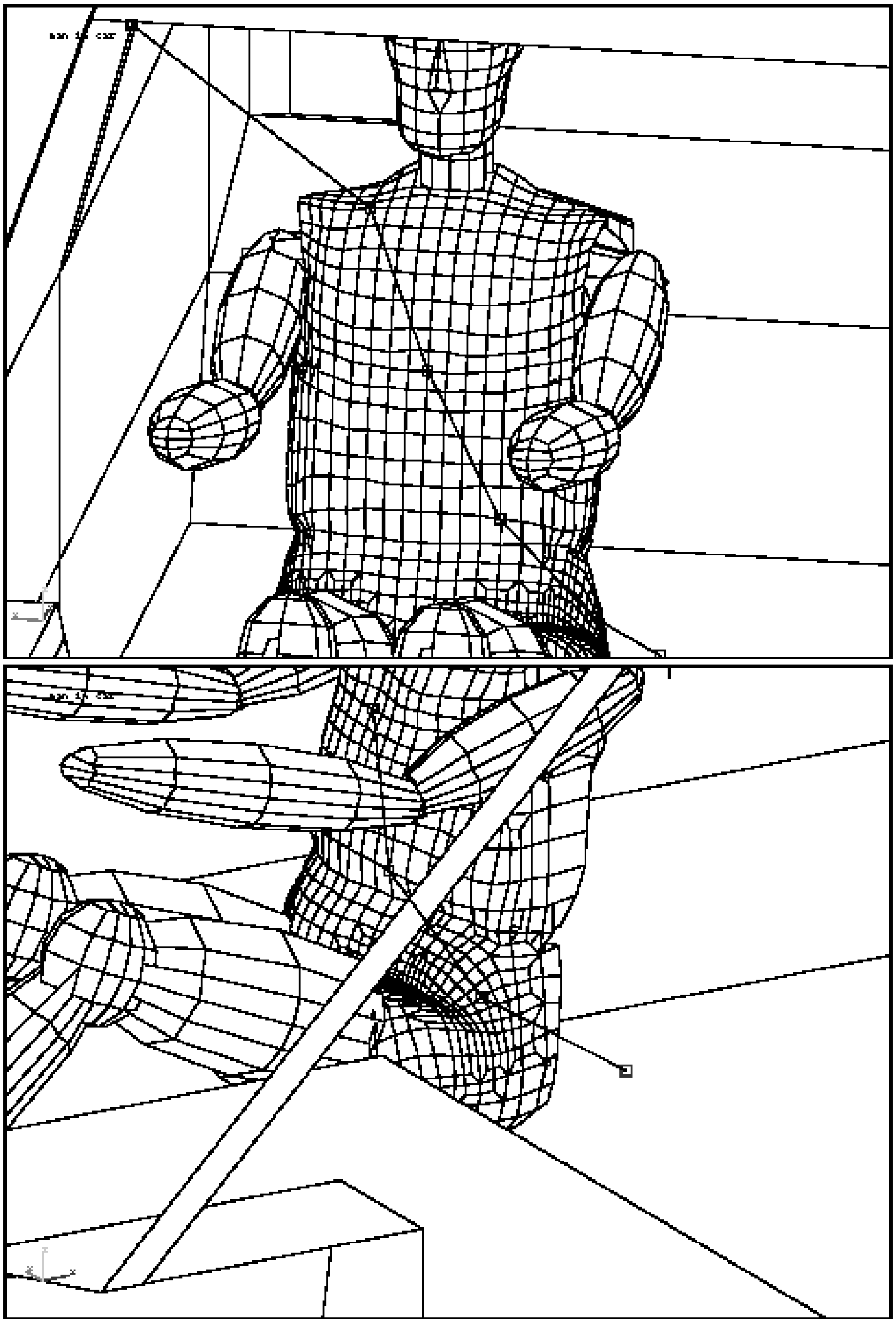
The mode was then swapped back to "node->node" for picking points across the pelvis round to the base of the B post:

```
 NODE->NODE        Mode is now node-node
 <pick node>
 <pick node>
 <pick node>
 <pick node>
       DONE        To finish picking nodes
 2600 700 300            Position on B pillar
DONE               To finish line definition
```

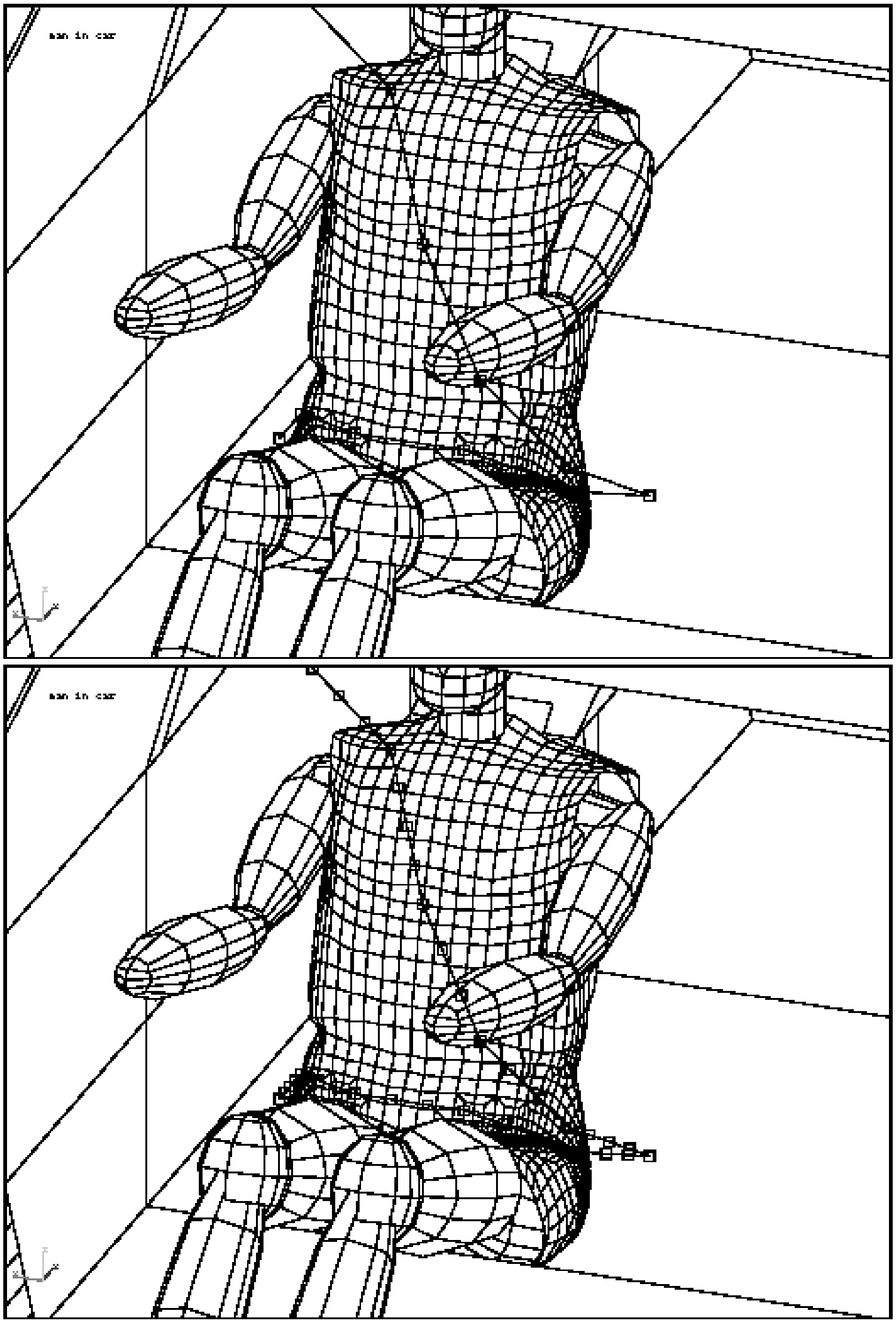
Subdivide the line into suitable belt-sized sections

Once the line is complete it must be sub-divided into (say) 50mm segments, since this is a suitable characteristic length for belt elements. You are prompted for this and given suitable choices. Figure A/30 shows basic and sub-divided line.

```
50.0              Select 50mm as suitable
```



Appendix III: Seat-belt positioning: Defining a crude line across dummy and pelvis. Location of D ring figure **A/29**



Appendix III: Seat-belt positioning. Final crude line definition, and its sub-division into 50mm segments. figure **A/30**

Choose suitable contact surface groups

For this dummy groups have already been defined in the "tree" file, see Appendix I. Only one group is suitable for the seat belt, and that is:

BELT_TO_DUMMY Choose just this group

It is useful to see what this represents, and indeed you may be curious to see what else is available. So you can "sketch" these groups using:

OPTIONS, SK_SURF To sketch contacts

Try sketching the "belt_to_dummy" group, and you will see that it covers neck, upper torso, mid torso and pelvis.

Tension the belt

When you have completed the above you will drop into the belt tensioning menu. The simplest thing to do is just to try it, using:

GO To start tensioning

If you try this you will get the message: **"WARNING: acute angle(s) found in line at unfixed point(s)."**

What this means is that the code has checked the angular difference between each line segment, and found that one or more is less than 90 degrees. This will be at the "D" ring position which we have forgotten to fix. So respond:

FIX To fix suspect point(s)

And use the mouse to click on the point on the "D" ring (behind the left side of the pelvis). It will be marked in green to show that it is fixed. Then type:

DONE To finish fixing
GO To try again

The belt will be dragged iteratively across the dummy until it reaches the convergence tolerance, or until it reaches a multiple of 25 iterations.

There is no "right" answer to a belt path, and you must judge whether the path achieved is acceptable or not.

In either event you can type:

GO	Continues iterating
CH_TOL	Changes convergence tol
DONE	To accept the shape
<other options>	

Section 5.2.6(d) et seq gives an explanation of the options available, and advice on what to do about problems.

In this case the author's belt converged after 21 iterations, and the initial (marked by squares) versus converged (marked by triangles) shapes are shown in figure A/31. So he used:

DONE	To accept the belt line

<u>Convert the line segments to belt elements</u>

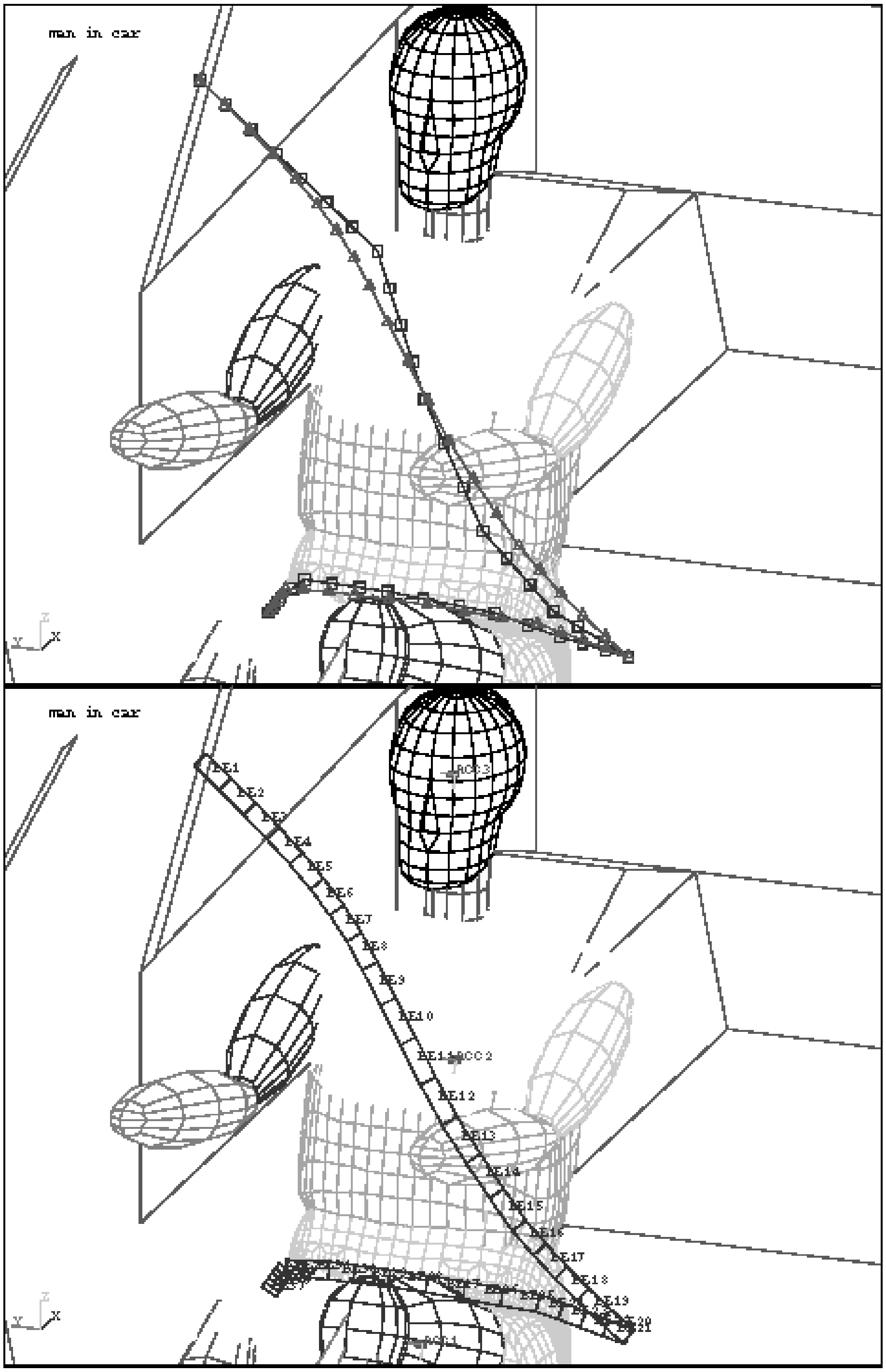
At present it is only possible to make "simple" belt elements, so use:

SIMPLE	Use simple elements
BMAT_1	Using belt matl #1

at this prompt to turn the line segments into seat-belt elements of belt material #1.

You will get messages about nodes on belts becoming slave nodes to the type 5 contacts on the dummy, and these contacts (which were defined as type 4 single surface) being converted to the master segments of a type 5.

And you have created a seat-belt. Use "sketch" in the **SEAT_BELTS** menu to draw the belt elements. The author's belt is shown in figure A/31.



Appendix III: Seat-belt positioning. Iterating to get belt shape, and conversion to seat-belt elements. figure **A/31**