

APPENDIX L: VDA Database

VDA surfaces describe the surface of geometric entities and are useful for the simulation of sheet forming problems. The German automobile and automotive supplier industry (VDA) has defined the VDA guidelines [VDA 1987] for a proper surface definition used for the exchange of surface data information. In LS-DYNA, this format can be read and used directly. Some files have to be provided for proper linkage to the motion of the correlation parts/materials in LS-DYNA.

Linking is performed via names. To these names surfaces are attached, which in turn can be linked together from many files externally to LS-DYNA. Thus, arbitrary surfaces can be provided by a preprocessor and then can be written to various files. The so-called VDA file given on the LS-DYNA execution line via `V = vda` contains references to all other files. It also contains several other parameters affecting the treatment in the contact subroutines; see below.

The procedure is as follows. If VDA surfaces are to be used, the file specified by `vda` must have the following form. The file is free formatted with blanks as delimiters. Note that the characters `"}"` and `"{"` must be separated from the other input by spaces or new lines. The `vda` file may contain any number of input file specifications of the form:

```
file afile bfile {
    alias definitions
}
alias definitions
followed by optional runtime parameters and a final end statement.
```

The file, `afile`, is optional, and if given must be the name of an ASCII input file formatted in accordance with the VDA Surface Interface Definitions as defined by the German automobile and automotive supply industry. `bfile` is required, and is the name of a binary VDA file. In a first run `afile` is given and `bfile` is created. In any further run, if the definitions have not changed, `afile` can be dropped and only `bfile` is needed. The purpose of `bfile` is that it allows for much faster initialization if the same VDA surfaces are to be used in a future LS-DYNA run.

If `afile` is given, `bfile` will always be created or overwritten. The alias definitions are used for linking to LS-DYNA and between the various surface definitions in the files defined by `afile` and `bfile`.

The alias definitions are of the form

```
alias name { el1 el2 ... eln }
```

where `name` is any string of up to 12 characters, and `el1,...,eln` are the names of VDA elements as specified in `afile`. The list of elements can be empty, in which case all the SURF and FACE VDA elements in `afile` will be used. Care should be taken to ensure that the alias `name` is unique, not only among the other aliases, but among the VDA element

APPENDIX L

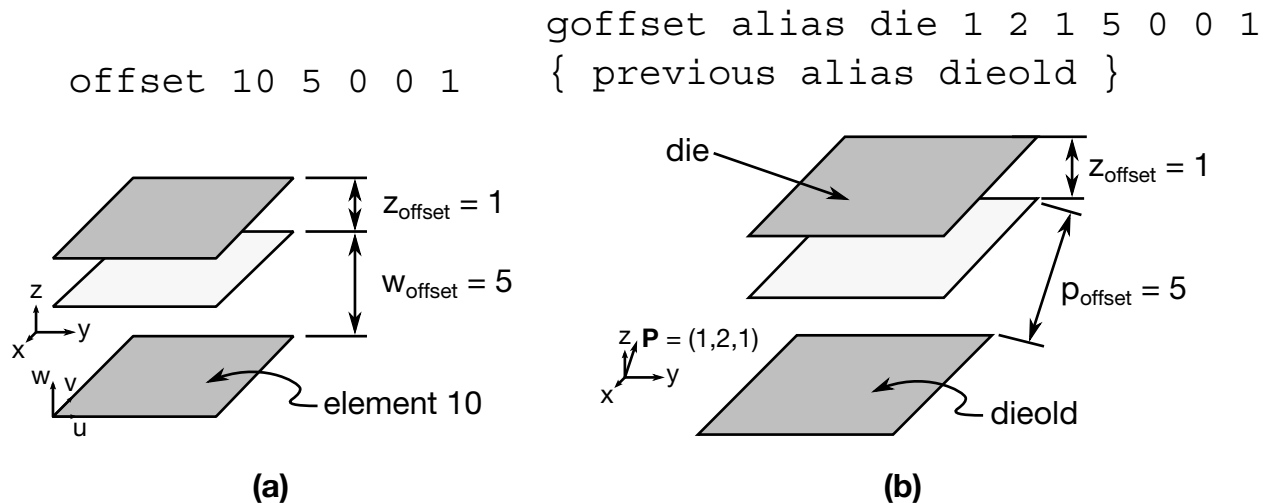


Figure 62-1. (a) a schematic illustration of offset version 1, and (b) is a schematic illustration of offset version 2.

names in **afile**. This collection of VDA elements can later be indicated by the alias **name**. In particular, **name** may appear in later alias definitions.

Often it is required that a punch or die be created by a simple offset. This can be achieved in the **vda** files in two ways, either on VDA elements directly, or on parts defined by aliases. This feature offers great capability in generating and using surface data information.

Offset Version 1

As an option, the keyword **offset** may appear in the alias list which allows a new surface to be created as a normal offset (plus translation) of a VDA element in the file. The keyword **offset** may be applied to VDA elements only, not aliases. The usage of **offset** follows the form

offset elem **normal x y z**

where **normal** is the amount to offset the surface along the normal direction, and **x,y,z** are the translations to be applied. The default normal direction is given by the cross product of the local **u** and **v** directions on the VDA surface, taken in that order. **normal** can be negative.

Offset Version 2

Frequently, it is convenient to create a new alias **name** by offsetting and translating an existing **name**. The keyword **goffset** provides this function:

goffset alias **name xc yc zc normal x y z** { previous **alias name** }

where **normal**, **x**, **y**, and **z** are defined as in the offset keyword. A reference point x_c , y_c , and z_c defines a point in space which determines the normal direction to the VDA surface, which is a vector from the origin to $P(x_c, y_c, z_c)$. See example below.

Finally, several parameters affecting the VDA surface iteration routines can be reset in the file **vda**. These parameters, and their default values in square brackets [], are:

- gap** [5.0] The maximum allowable surface gap to be filled in during the iterations. Points following the surface will effectively extend the edges of surfaces if necessary to keep them from falling through cracks in the surface smaller than this. This number should be set as small as possible while still allowing correct results. In particular, if your VDA surfaces are well formed (having no gaps), this parameter can be set to 0.0. The default value is 5.0.
- track** [2.0] A point must be within this distance of contact to be continually tracked. When a point not being tracked comes close to a surface, a global search is performed to find the near surface point. While a point is being tracked, iterations are performed every cycle. These iterations are much faster, but if the point is far away it is faster to occasionally do the global search. The default value is 2.0.
- track2** [5.0] Every VDA surface is surrounded by a bounding box. When a global search needs to be performed but the distance from a point to this box is $>$ **track2**, the actual global search is not performed. This will require another global search to be performed sooner than if the actual distance to the surface were known, but also allows many global searches to be skipped. The default value is 5.0.
- ntrack** [4] The number of VDA surfaces for which each point maintains actual distance information. A global lower bound on distance is maintained for all remaining surfaces. Whenever the point moves far enough to violate this global lower bound, all VDA surfaces must have the global search performed for them. Hence, this parameter should be set to the maximum number of surfaces that any point can be expected to be near at one time (the largest number of surfaces that come together at one point). Setting **ntrack** higher will require more memory but result in faster execution. If **ntrack** is too low, performance may be unacceptably slow. The default value is 4.0.

APPENDIX L

- toroid** [.01] Any surface with opposing edges which are within distance [t] of each other is assumed to be cylindrical. Contacts occurring on one edge can pass to the adjacent edge. The default value is 0.01.
- converge** [.01] When surface iterations are performed to locate the near point, iteration is continued until convergence is detected to within this distance (all VDA coordinates are in mm). The default value is 0.01.
- iterate** [8] Maximum number of surface iterations allowed. Since points being tracked are checked every cycle, if convergence fails it will be tried again next cycle, so setting this parameter high does not necessarily help much. On the other hand, a point converging to a crease in the VDA surface (a crease between patches with discontinuous derivative, for example) may bounce back and forth between patches up to this many times, without actually moving. Hence, this value should not be too large. The default value is 8.
- el_size** [t mx mn]
- Controls the generation of elements where:
- t = surface tolerance for mesh generation,
 - mx = maximum element size to generate,
 - mn = minimum element size to generate.
- The default values are [0.25 100. 1.0]
- aspect** [s1 s2] Controls the generation of elements where:
- s1 = maximum difference in aspect ratio between elements generated in neighboring VDA patches,
 - s2 = maximum aspect ratio for any generated element.
- The default values are [1.5 4.0]
- cp_space** [10] Determines the spacing around the boundaries of parts at which the size of elements is controlled. In the interior of the part, the element size is a weighted function of these control points as well as additional control points in the interior of the region. If there are too few control points around the boundary, elements generated along or near straight boundaries, but between control points, may be too small. The default value is 10.
- meshonly** The existence of this keyword causes LS-DYNA to generate a file containing the mesh for the VDA surfaces and then terminate.
- onepatch** The existence of this keyword causes LS-DYNA to generate a single element on each VDA patch.

somepatch [n] Like onepatch, but generates an element for 1 out of every [n] patches.

Example for file V = **vda**. It contains the following data:

```
file vda1 vda1.bin {
    alias die {
        sur0001
        sur0003
        offset fce0006 1.5 0 0 120
    }
    alias holder1 { sur008 }
}
file vda2 vda2.bin {
    alias holder2 { sur003 }
}
alias holder { holder1 holder2 }
ntrack 6
gap 0.5
end
```

Explanation:

vda1	This file contains the surfaces/face elements sur0001,sur0003, fce0006, and sur0008.
alias die face	Combines the surface/face elements sur0001, sur0003, and the offsetted fce0006 to a global surface.
alias holder1	Defines the surface/face element sur0008 as holder1.
vda2	This file contains the surface/face element sur0003.
alias holder2	Defines the surface/face element sur0003 as holder2.
alias holder	Combines the surfaces holder1 and holder2 into a combined surface holder.
ntrack 6	For each point the actual distances to 6 VDA surfaces are maintained.
gap 0.5	Surface gaps of 0.5mm or less are filled.
end	Closes reading of this file.

