ESP Training Sessions 3 & 4 (June 11, 2021)
Muddy Cards

Can you give us the demonstration file that you used at the end of session 4?

```
# muddy1
# written by John Dannenhoffer

CYLINDER  -5  0  0 \
          +5  0  0  0.5
STORE     cylinder  0  1

TORUS      0  0  0   0  0  1  3.0  1.0
STORE     torus     0  1

# get the left-most Body
#--- RESTORE    cylinder
#--- RESTORE    torus
#--- SUBTRACT   xmin   1
#--- SELECT     FACE
#--- ATTRIBUTE  _color $red

# get the rite-most intersection
#--- RESTORE    cylinder
#--- RESTORE    torus
#--- INTERSECT  xmax  99
#--- SELECT     FACE
#--- ATTRIBUTE  _color $green

END
```

Do we need to use ATTRIBUTE in order to SUBTRACT?

Attributes are totally independent of the stack. They are simply properties of a Body, Face, Edge, or Node.

Is it possible to assign different densities to different Bodys?

The @xcg and @Ixx properties all assume that the density is one (that is, xcg is actually the x-centroid). If you wanted to find the xcg and Ixx of multiple Bodys with different densities, you would need to do the CG or centroid calculations yourself, using for example, the parallel axis theorem.

Can you toggle between perspective and parallel view to more easily check if things line up as intended?

No. The choice can only be made with the command line option -onormal

Are there plans to be able to dump collada (dae) files?

Not at this time, but we can talk about it if you wish

Can you use multiple MARK commands, or is the previous MARK removed?

There can be as many Marks on the stack as you need. Operations that use the Mark (such as RULE) pop the Mark off the stack in the same way as it pops Bodys off the stack.

What is the purpose of doing a STORE immediately followed by a RESTORE? Does it just allow for a second RESTORE later on? Seems like it is that the same as Cut and Paste, so can Paste again later.

If you want to duplicate the Body on the top of the stack, you could do:

```
STORE   foo
RESTORE foo
RESTORE foo
```

Or you could do:

```
STORE   foo  0  1
RESTORE foo
```

Or (starting in v1.19), you could do:

```
RESTORE .
```

The only benefit of the first two is that you now have the storage "foo 0" that you can RESTORE elsewhere in your script.

Is @xmin, @xmax, etc of the body on top of the stack or all of all bodies?

The at-parameters are computed for all currently items currently in the SelectList. See session 8 for details

Is there are difference between RULE and BLEND if there are only 2 sections (i.e. if the wing problem was made with just the root and tip and then mirrored)?

No, with two sections they produce the same result

Do local variables have a scope? (i.e. if set within a PATBEG/PATEND loop, are they accessible outside the loop.

The scopes in ESP are for the .csm or .udc file, unless the UDC is an include-type UDC. Variables with PATBEG/PATEND are accessible outside the loop. See session 5 for details.

In calculations, are typical order of operations handled correctly or just left to right? Does 2+3*4 = 14 or 24?

ESP uses the conventional rules, so 2+3*4 is the same as 2+(3*4), which evaluates to 14

Could you please further review what the significance of the declarations CFGPMTR, CONPMTR, and OUTPMTR are as opposed to DESPMTR? What instances should we use either of these?

See session4, slide 14 for a summary.

DESPMTRs can be (re-)set before a build, such as through CAPS or an optimizer. ESP can compute sensitivities with respect to

a DESPMTR

CFGPMTRs can be (re-)set before a build, such as through CAPS or an optimizer. The only difference between a CFGPMTR and DESPMTR is that ESP cannot compute the sensitives with respect to a CFGPMTR. CFGPMTRs are typically used for things such as the number of ribs in a wing or a switch to determine if only a half or a whole airplane is to be generated.

CONPMTRs are used to define "constants". For example, you might want to define the acceleration of gravity

```
CONPMTR g 9.81
```

By having this as a CONPMTR, you will get an error if you accidentally try to give "g" a new value. CONPMTRs can either be defined with a CONPMTR statement (as above) or in a dictionary, which is loaded with the -dict command-line option to serveCSM

OUTPMTRs are LocalVariables that can be set and re-set during the build process. What distinguishes an OUTPMTR from an ordinary LocalVariable is that OUTPMTRs are see-able outside ESP (such as in CAPS)

So what is the benefit of using a configuration parameter instead of a design parameter? It sounds like config is just a subset of the functionality.

Yes, CFGPMTRs are essentially the same as DESPMTRs, except you cannot get the sensitivity w.r.t a CFGPMTR

Are configuration parameters mainly for discrete parameters or are there other typical uses for it?

Typically they are used for discrete parameters, but can used for any parameter for which computing the sensitivity does not make sense

Please explain again the various continuities in BLEND.

Image you have the following on the stack:

```
mark xsect1 xsect2 xsect3 xsect4
```

BLEND will generate a Body that is C2 (continuous second derivative - or curvature) at xsect2 and xsect3

Image you have the following on the stack:

```
mark xsect1 xsect2 xsect2 xsect3 xsect4
```

BLEND will generate a Body that is C1 (continuous first derivative - or slope) at xsect2 and C2 at xsect3

Image you have the following on the stack:

```
mark xsect1 xsect2 xsect2 xsect2 xsect3 xsect4
```

BLEND will generate a Body that is C0 (continuous zeroth derivative - or value) at xsect2 and C2 at xsect3

Please explain again how nose treatment works in BLEND.

If the first (last) is a POINT and begList (endList) contains 8 elements, then a rounded beginning (ending) will be created by BLEND. The meaning of begList (endList) is:

```
rad1;dx1;dy1;dz1;rad2;dx2;dy2;dz2
```

rad1 is the radius of curvature when looking in the dx1;dy1;dz1 direction. rad2 is the radius of curvature when looking in the dx2;dy2;dz2 direction

What is the directory for simple uses of ESP Commands

EngSketchPad/data/basic (LINUX/OSX)
EngSketchPad\data\basic (Windows)

How do you know what side is the beginning or end of the line to put points on it?

If you generate it via a Sketch, it goes in the direction that you defined it

You can look at the orientation [Ori] of an edge in the ESP viewer.

If in waffle, line always goes from <pname1> to <pname2>

Where can I find the user-defined primitives (box,waffle...)? In which folder?

They are in EngSketchPad/lib, but you do not really need to know this to use them

Where can I find the user-defined components (gen_rot,..)? In which folder?

They are in EngSketchPad/udc

Can PATBEG/PATEND be nested patterns?

They can be nested 20 levels deep

Is "PI" (3.14159...) a pre-defined constant parameter?

No, use the pi(1) function

Can DIMENSION be defined by a CFGPMTR or CONPMTR?

```
CONPMTR   numRows 3
CFGPMTR   numCols 2
DIMENSION array   numRows numCols
SET       array   "5;2"
END
```

produces:

```
array = 5, 2, 2;
        2, 2, 2
```

BTW: this also works for a DESPMTRs

What is the value of nint(3.5) and nint(-3.5)?

nint(x) is computed by:
if x>0, integer part of (x+0.5)
if x<0, integer part of (x-0.5)
if x=0, 0

So, nint(3.5)=4 and nint(-3.5)=-4

Can you use val2str within a pattern to name each body made in the loop when storing it using the iteration value? For example: store body_val2str(i)?

Actually you would need: STORE !$body_+val2str(i) since the first argument to STORE is an implicit string

Do all the newly-created Edges in a RULE or BLEND have to be unique? That is, can I generate a prism by making Edges on consecutive xsects the same.

RULE and BLEND require that the xsects be unique (except for a repeated xsect to get C1 or C0 continuity). If you want to create a prism, pyramid, or tetrahedron, consider using UDPRIM poly instead

Can a csm file include another file, for example to better organize a large model?

Yes, use an include-type UDC.

Explain again what the colon (:) in a parameter name means

It is used only for organizing the DesignParameter list in ESP. The computations on parameters with or without a colon are exactly the same.

Please explain @-parameters again

They are created whenever a Body is created or a SELECT statement is executed. See session 8