

Engineering Sketch Pad (ESP)



Advanced User Training Session 04 Advanced Surface Generation

John F. Dannenhoffer, III

john@geocentrictech.com

Geocentric Technologies LLC

updated for v1.29+

- Using BLEND
 - C-2, C-1, and C-0 continuity
 - transition regions
 - `noselist` and `taillist`
- Aligning components
 - EVALUATE CLOSE
 - OPTBEG and OPTEND
- Using `offsetExp` and `flendExp`
- Homework exercises

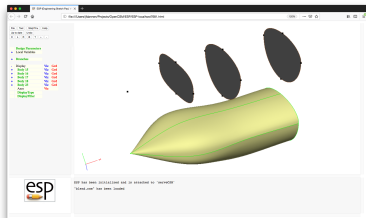
- RULE and BLEND require that all Xsects have the same number of Segments, ordered in the same way
 - new Faces are made by combining all the first Segments, ...
 - the `.multiNode` attribute can override for RULE
- BLEND allows user-selectable continuity in blend direction
 - C2 - curvature continuity (the default)
 - C1 - slope continuity (obtained with Xsect repeated once)
 - C0 - value continuity (obtained with Xsect repeated twice)
- Xsects can be automatically reordered to help eliminate twist by setting `reorder` to a non-zero value
 - positive to start from first Xsect
 - negative to start from last Xsect
- Users can manually reorder Xsects with the `REORDER` command (applied to a Xsect)
 - Reordering only changes the order of Segments, not their shapes

- If all Xsects are NodeBodys
 - a WireBody is produced
- If the first and last Xsects are both WireBodys
 - a SheetBody is produced that is open on both ends
- If the first or last Xsect is a WireBody
 - a SheetBody is produced that is open on one end and closed on the other
- Otherwise
 - a SolidBody is produced



Grown Primitive — BLEND

Note: Original Xsects (SheetBodys) and result of BLEND are shown



```
# blend
```

```
MARK
```

```
POINT 0 0 0
```

```
UDPRIM supell rx 2 ry_n 1 ry_s 1 n 3
```

```
ROTATEY 90 0 0
```

```
TRANSLATE 3 0 0
```

```
UDPRIM supell rx 2 ry_n 1 ry_s 2
```

```
ROTATEY 90 0 0
```

```
TRANSLATE 6 0 0
```

```
UDPRIM supell rx 2 ry_n 1 ry_s 2
```

```
ROTATEY 90 0 0
```

```
TRANSLATE 10 0 0
```

```
GROUP
```

```
STORE sections
```

```
RESTORE sections
```

```
TRANSLATE 0 4 0
```

```
MARK
```

```
RESTORE sections
```

```
BLEND
```

```
END
```

● Face-order on later slide

- (1) first Xsect (or empty if NodeBody or WireBody)
- (2) last Xsect (or empty if NodeBody or WireBody)
- (3) Face from first Xsect Edge between first and second Xsects
- (4) Face from first Xsect Edge between second and third Xsects
- ...
- (n) Face from second Xsect Edge between first and second Xsects
- ...



BLEND Continuity (1)

```
# blendCOC1C2
```

```
# original Xsects (top left)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
GROUP
```

```
TRANSLATE -3 +1 0
```

```
# Body with C0 at second Xsect (top rite)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
BLEND
```

```
TRANSLATE +3 +1 0
```

```
# Body with C1 at second Xsect (bottom rite)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
BLEND
```

```
TRANSLATE -3 -1 0
```

```
# Body with C2 at second Xsect (bottom left)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

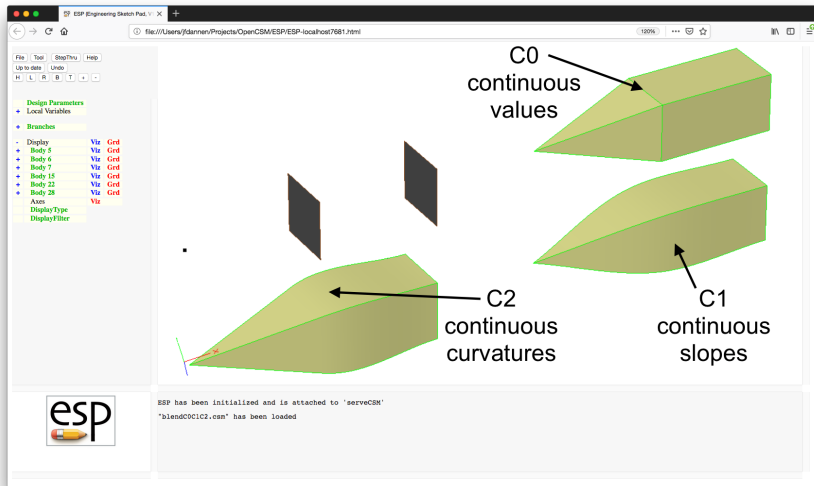
```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
BLEND
```

```
TRANSLATE +3 -1 0
```

```
END
```



```
# blend with two C1s (using slopes between the C1s)
```

```
MARK
```

```
BOX      -2  -2  -2  0  4  4
```

```
BOX      -1  -1  -1  0  2  2
```

```
ATTRIBUTE .C1side $fwd
```

```
RESTORE .
```

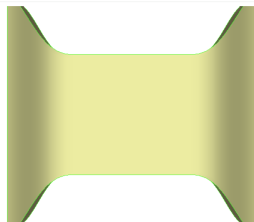
```
BOX      1  -1  -1  0  2  2
```

```
ATTRIBUTE .C1side $rev
```

```
RESTORE .
```

```
BOX      2  -2  -2  0  4  4
```

```
BLEND
```



blend with two C1s (using slopes outside the C1s)

MARK

```
BOX      -2  -2  -2  0  4  4
```

```
BOX      -1  -1  -1  0  2  2
```

```
ATTRIBUTE .C1side $rev
```

```
RESTORE .
```

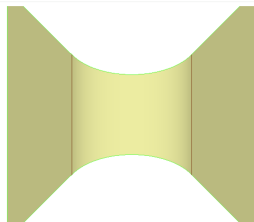
```
BOX      1  -1  -1  0  2  2
```

```
ATTRIBUTE .C1side $fwd
```

```
RESTORE .
```

```
BOX      2  -2  -2  0  4  4
```

BLEND





BLEND Nose/Tail Treatment (1)

```
# blendC0C1C2
```

```
# original Xsects (top left)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
GROUP
```

```
TRANSLATE -3 +1 0
```

```
# Body with pointed nose (top rite)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
BLEND
```

```
TRANSLATE +3 +1 0
```

```
# Body with slightly rounded nose (bottom left)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

```
TRANSLATE +2 0 0
```

```
BLEND "0.1; 0;1;0; 0.1; 0;0;1"
```

```
TRANSLATE -3 -1 0
```

```
# Body with rounded nose (bottom rite)
```

```
MARK
```

```
POINT -2 0 0
```

```
UDPRIM box dy 1 dz 1
```

```
UDPRIM box dy 1 dz 1
```

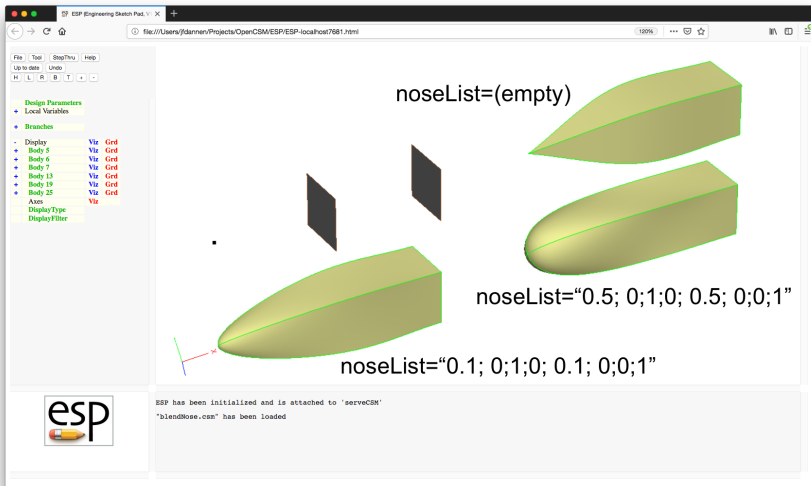
```
TRANSLATE +2 0 0
```

```
BLEND "0.5; 0;1;0; 0.5; 0;0;1"
```

```
TRANSLATE +3 -1 0
```

```
END
```

BLEND Nose/Tail Treatment (2)



- If the first Xsect is a SheetBody with 2 or 3 Edges and the `begList` contains 2 entries:
 - `begList[1] = -1` (to distinguish this from rounding)
 - `begList[2] =` the aspect ratio of an approximate ellipse that spans between the first and second Xsect Edge
- The same applies to the last Xsect and `endList`



- Call is: EVALUATE CLOSE ibody1 itype1 ient1 ibody2 itype2 ient2
 - ibody1 is number of first Body
 - itype1 identifies the entity in ibody1
 - if 0, ient1 is Node number
 - if 1, ient1 is Edge number
 - if 2, ient1 is Face number
 - ibody2, itype2 and ient2 are similarly defined
- Returned values are:
 - @edata[1] is minimum distance between the entities
 - @edata[2] and @edata[3] are parametric coordinates of closest point in ient1 (or zero)
 - @edata[4] through @edata[6] are the coordinates of closest point in ient1
 - @edata[7] through @edata[11] are similar values for ient2

OPTBEG \$objvar \$varList lbounds ubounds
use: starts an Optimizer Block
pops: -
pushes: -
notes: Solver must not be open
 Optimizer must not be open
 opens an Optimizer Block
 objval is the LOCALVAR to be minimized
 varList is a list of semicolon-separated LOCALVARs
 varList must end with a semicolon
 lbounds is a vector of lower bounds
 ubounds is a vector of upper bounds
 cannot be followed by ATTRIBUTE or CSYSTEM

```
DESPMTR    xcent    1.5
DESPMTR    ycent    -.5

# create necessary local parameters (values do not matter)
SET        x        0
SET        y        0
SET        obj       0

# optimizer loop
OPTBEG     obj x;y;  -2  +2
    SET     obj  5*(x-xcent)^2+(y-ycent)^2
OPTEND

# create a point at the minimum
POINT     x y  0
```

- Two-step process:
 - ① identify and/or generate Curves that identify the extents of the FLEND
 - ② create the FLEND between the two Curves

```
UDPRIM offsetExp
input Body(s)
  * SolidBody
output Body(s)
  * scribed SolidBody
input arguments (specified as name/value pairs):
```

argname	description	default
\$faceAttr	attribute name or name=value on Faces on which the offset can be made	
\$edgeAttr	attribute name or name=value for Edges from which the offset will be created	
dist	the approximate distance between the offset Edges and the Edges marked with edgeAttr	
\$outAttr	name of attribute to be put on offset Edges and Faces between the offset Edges and the Edges with the edgeAttr attribute	

output arguments:

argname	description
	-NONE-

usage:

- * the Body produced is the input Body scribed by the offset Edges (with Edges and Faces marked with outAttr)
- * triggers finite difference sensitivities
- * this is an experimental UDF, which means that results from future versions will likely NOT be backward compatible with the current version

```
UDPRIM flendExp
input Body(s)
  * SolidBody
output Body(s)
  * new SolidBody
input arguments (specified as name/value pairs):
```

argname	description	default
\$side1attr	\$side1attr attribute name or name;value on the family of Edges on one side of the flend and on the Faces adjacent to the Edges that are to be replaced by the flend faces	
\$side2attr	attribute name or name;value on the family of Edges on the other side of the flend and on the Faces adjacent to the Edges that are to be replaced by the flend Faces	
wght1	value of the curvature weight function adjacent to the Edges marked with side1attr	1
wght2	value of the curvature weight function adjacent to the Edges marked with side2attr	1
\$outattr	attribute to be put on the new flend Faces and the Edges between flend Faces	
mateNodes	set to 1 to use only existing Nodes	0

output arguments:

argname	description
	-NONE-

usage:

- * `wght1=wght2=1` minimizes the integral of the curvature across the flend
- * `wght* > 1` means that the curvature weighting is more at an end than at the middle of the flend curves
- * `wght* < 1` means that the curvature weighting is less at an end than at the middle of the flend curves
- * a Face can be marked with both `side1attr` and `side2attr`
- * all Faces between Edges in `side1attr` and `side2attr` must be marked with either `side1attr` or `side2attr`
- * if `mateNodes=1`, the number of Edges marked with `side1attr` must be the same as the number of Edges marked with `side2attr`
- * this is an experimental UDF, which means that results from future versions will likely NOT be backward compatible with the current version

- Start with `exercises/session04/wing.csm`
- Add a DESPMTR called `:tipfac`
 - if `:tipfac = 0`, generate the original wing
 - else, add a rounded tip with the given `:tipfac`
- Try with `:tipfac` set to 3

- Start with `exercises/session04/wing.csm`
- Add a DESPMTR called `:dytrans`
 - if `:dytrans = 0`, generate the original wing
 - else, add a smooth transition a distance $\pm$: `dytrans` from the break
- Try with `:dytrans` set to 2

- Start with `exercises/session04/roundedlip.csm`
- Add a rounded lip at the inlet

- Start with `exercises/session04/align.csm`
- Add a `OPTBEG/OPTEND` block to move the wing such that its leading- and trailing-edges match the seam on the side of the fuselage

- Start with `exercises/session04/align.csm`
- Add a `OPTBEG/OPTEND` block to modify the fuselage shape such that the wing's leading- and trailing-edges match the seam on the side of the fuselage

- Start with `exercises/session04/flend.csm`
- Add the `wingDist` DESPMTR and set to 0.30
- Generate an offset curve on the wing that is `wingDist` away from the intersection curve
- Add the `fuseDist` DESPMTR and set to 0.22
- Generate an offset curve on the fuselage that is `fuseDist` away from the intersection curve
- Create a flend between the offset curves