

(UDPs/UDFs — continued from other side)

```
radwaf      ysize zsize nspoke xframe[]
sew         $filename toler bodynum
stag        rad1 beta1 gama1 rad2 beta2 gama2 ...
           ... alfa xfrnt xrear
stiffener   beg[] end[] depth angle
supell      rx rx_w rx_e ry ry_s ry_n n_nw n_e ...
           ... n_s n_n n_sw n_se n_nw n_ne offset nquad
waffle      depth segments[] $filename progress
```

Built-in Functions

General

```
pi(x)
min(x,y)
max(x,y)
sqrt(x)
abs(x)
int(x)
nint(x)
ceil(x)
floor(x)
mod(a,b)
sign(test)
exp(x)
log(x)
```

Trigonometric

```
log10(x)
sin(x)
sind(x)
asin(x)
asind(x)
cos(x)
cosd(x)
acos(x)
acosd(x)
tan(x)
tand(x)
atan(x)
atand(x)
atan2(y,x)
atan2d(y,x)
hypot(x,y)
hypot3(x,y,z)
```

Sketch utilities

```
incline(xa,ya,dab,xb,yb)
Xcent(xa,ya,dab,xb,yb)
Ycent(xa,ya,dab,xb,yb)
Xmidl(xa,ya,dab,xb,yb)
Ymidl(xa,ya,dab,xb,yb)
seglen(xa,ya,dab,xb,yb)
radius(xa,ya,dab,xb,yb)
sweep(xa,ya,dab,xb,yb)
turnang(xa,ya,dab,xb,yb,dbc,xc,yc)
dip(xa,ya,xb,yb,rad)
smallang(x)
```

Conversions

```
val2str(num,digits)
str2val(string)
findstr(str1,str2)
slice(str,ibeg,iend)
path($pwd) or path($csm) or path($root) or path($file)
```

Logic

```
ifzero(test,ifTrue,ifFalse)
ifpos(test,ifTrue,ifFalse)
ifneg(test,ifTrue,ifFalse)
ifmatch(str,pat,ifTrue,ifFalse)
ifnan(test,ifTrue,ifFalse)
```

Dot-suffixes

```
x.nrow      number of rows in x or 0 if a string
x.ncol      number of columns in x or 0 if a string
x.size      number of elements in x (=x.nrow*x.ncol) or len of str x
x.sum       sum of elements in x
x.norm      L2-norm (RMS) of elements in x
x.min       minimum value in x
x.max       maximum value in x
```

Character Set

#	hash	introduces comment
"	quotes	ignore spaces until following "
\	backslash	ignore this and following characters and concatenate next line
<space>	space	separates arguments in .csm file (except between " and ")
0-9		digits used in numbers, names, and strings
A-Z a-z		letters used in names and strings
_ : @		characters used in names and strings
? % =		characters used in strings
.	period	decimal separator (used in numbers), introduces dot-suffixes (in names)
,	comma	separates function arguments and row/column in subscripts
;	semicolon	multi-value item separator
()	parentheses	groups expressions and function arguments
[]	brackets	specifies subscripts in form [row,column] or [index]
{ } < >		characters used in strings
+ - * / ^		arithmetic operators
\$	dollar	as first character, introduces a string that is terminated by end-of-line or un-escaped plus, comma, or open-bracket
@	at-sign	as first character, introduces @-parameters
'	apostrophe	used to escape comma, plus, or open-bracket within strings
!	exclamation	if first character of implicit string, ignore \$! and treat as an expression
	bar	cannot be used (reserved for OpenCSM internals)
~	tilde	cannot be used (reserved for OpenCSM internals)
&	ampersand	cannot be used (reserved for OpenCSM internals)

ESP User Interface

Keypress (if not caught by browser)

ctrl-h -or- <Home>	initial view (or H button)
ctrl-f	front view
ctrl-l	leftside view (or L button)
ctrl-r	riteside view (or R button)
ctrl-t	top view (or T button)
ctrl-b	bottom view (or B button)
ctrl-i -or- <PgUp>	zoom in (or + button)
ctrl-o -or- <PgDn>	zoom out (or - button)
<Left>	rotate or xlate (in flying mode) left
<Rite>	rotate or xlate (in flying mode) rite
<Up>	rotate or xlate (in flying mode) up
<Down>	rotate or xlate (in flying mode) down
>	save view
<	recall view
ctrl-> -or- .	save view to file
ctrl-< -or- ,	read view from file
^ -or- 6	query object at cursor
@ -or- 2	get coords. @ cursor
v	toggle Viz (visability) at cursor
g	toggle Grd (grid) at cursor
t	toggle Trn (transparency) at cursor
o	toggle Ori (orientation) at cursor
A	add Attribute at cursor
* -or- 8	center view @ cursor
!	toggle flying mode