

Costruzione CILINDRO o CONO da punti misurati nel QUINDOS

La costruzione del cilindro e del cono da punti misurati (per esempio dai punti di due o più cerchi misurati su diversi sezioni) presenta alcuni punti critici, che se non si conoscono bene possono impedire l'utente di eseguire tale operazione, che sembra così semplice.

Inanzitutto cerchiamo di immaginare cosa deve fare il software per riuscire a completare questa operazione. Come partenza del calcolo noi abbiamo punti di misura sparsi per aria. Questi punti possono appartenere a qualche cerchio o se sono stati presi sulla generatrice dell'elemento, allora appartengono ad una asse. Si tratta sempre dei punti misurati e quindi punti "centro tastatore". La operazione che sembra più logica è di raggruppare tutti i punti in un elemento tramite COLAPT, e far calcolare un cilindro o cono da questi punti (nello stesso COLAPT o con un comando seguente BLDCYL o BLDCON). Il raggruppamento è una semplice operazione e non è critica per niente. Il problema più difficile in questo caso è l'esecuzione del calcolo. Ricordiamoci che sia il cilindro, che il cono sono elementi tredimensionali e matematicamente il risultato del calcolo di questi elementi dai punti dipende da come inizia il calcolo. Questo è il motivo che esiste una certa sequenza predefinita quando si misura il cilindro oppure il cono in manuale – basta sbagliare questa sequenza e poi sbaglia anche il calcolo!!! Ecco perchè raggruppando i punti che sono in una sequenza qualsiasi non ci permette ancora di avere una certezza su come avviene il calcolo. Importante è capire questo fatto che si può riassumere così: Il solo insieme dei punti sparsi nel volume tredimensionale non è sufficiente per eseguire il calcolo di un cilindro o un cono. Bisogna in qualche maniera fornire l'informazione almeno approssimativa che permetterà al algoritmo del calcolo di riuscire.

Il problema quindi è di informare QUINDOS quale è la posizione approssimativa del elemento da calcolare. Diciamo subito che QUINDOS usa il sottotipo del elemento NOM per informarsi su questo. Ecco come si presenta la maschera del sottotipo NOM di un cerchio misurato a mano:

E L E M E N T D A T A					
ELEMENT	-TYPE	=	CIR	COORD. -SYSTEM	= CSY (0)
				EVAL. -SYSTEM	= (0)
PROJECTION-TYPE	=	SM	INT. / EXT.	=	I
RAD. -CORR. TYPE	=		RAD. -CORR. VALUE	=	3
EVAL. -TYPE	=	Y	SIGMA-DISPLACEM.	=	0.000
CALCUL. -TYPE	=	CIR	CODE	=	
COORDINATES	X =	0.000	Y =	0.000	Z = 0.000
SPACE-ALIGNMENT	U =	0.000	V =	0.000	W = 1.000
PLANE-ALIGNMENT	L =	1.000	M =	0.000	N = 0.000
ABSOLUTE-VALUES	A =	3.012	B =	0.000	C = 0.000
	D =	0.000	E =	0.000	F = 0.000
PROJECTION-PLANE	X =	0.000	Y =	0.000	Z = -272.217
	U =	0.000	V =	0.000	W = 1.000
RAD. -CORR. DIREC.	U =	1.000	V =	0.000	W = 0.000
ADD. PARAMETER	1 =	0.100	2 =	4.000	3 = 0.000
	4 =	0.000	5 =	0.000	6 = 0.000
ITERATION START VALUE [ITERATION CONTROL]					
X =	0.000	[]	Y =	0.000	[]
U =	0.000	[]	V =	0.000	[]
A =	0.000	[]	B =	0.000	[]
			Z =	0.000	[]
			W =	0.000	[]
			C =	0.000	[]

Notare che la parte ITERATION START VALUE (Valori iniziali d'iterazione) è praticamente vuota. Sappiamo benissimo che per un cerchio per esempio no esiste una certa sequenza di presa punti – questo è perché l'algoritmo per il cerchio non necessita' di una informazione sulla sua posizione approssimativa nello spazio. Non è però lo stesso per il cono e per il cilindro. La sequenza fissa della presa punti per un cono (3 su una circonferenza + 3 su un'altra circonferenza) permette al sistema di compilare la parte dei valori iniziali d'iterazione con il valore della direzione approssimativa del cono ricavando questo valore dal calcolo di un'asse che passa tra i centri dei 2 cerchi che QUINDOS si calcola da questa serie di 6 tastate. Adesso è chiaro perché sbagliando la sequenza iniziale può rovinare la procedura del calcolo anche se dopo si prendono tanti altri punti. Torniamo però di nuovo al nostro caso e vediamo adesso su un esempio come si risolve il problema. Supponiamo che sono stati misurati in scansione 2 sezioni di un cilindro – CER(1) e CER(2). L'asse del cilindro è approssimativamente nella direzione X del sistema di coordinate.

MECIR (NAM=CER(1))

MECIR (NAM=CER(2))

Il raggruppamento dei punti avviene tramite COLAPT

COLAPT (NAM=CIL, PTS=(CER(1), CER(2))

A questo punto l'esecuzione del BLDCYL può dare risultati davvero strani perché essendo un nuovo elemento, CIL non ha nessun valore impostato nella parte dei valori iniziali d'iterazione, che come abbiamo visto sono fondamentali per il calcolo. Quindi prima di eseguire BLDCYL dobbiamo assicurarci che i campi dei ITERATION START VALUE sono compilati in modo opportuno. Questo si può fare in diversi modi... Per esempio si può aprire il sottotipo NOM del elemento CIL e compilare i campi per la direzione dell'elemento U, V, W con i valori approssimativi e indicando con C che si tratta del coseno direttore nei campi accanto a U, V e W.

E L E M E N T D A T A						
ELEMENT	-TYPE	=		COORD.-SYSTEM	=	CSY (0)
				EVAL.-SYSTEM	=	(0)
PROJECTION	-TYPE	=		INT. / EXT.	=	
RAD.-CORR.	TYPE	=		RAD.-CORR. VALUE	=	
EVAL.	-TYPE	=		SIGMA-DISPLACEM.	=	0.000
CALCUL.	-TYPE	=		CODE	=	
COORDINATES	X	=	0.000	Y	=	0.000
				Z	=	0.000
SPACE-ALIGNMENT	U	=	0.000	V	=	0.000
				W	=	1.000
PLANE-ALIGNMENT	L	=	1.000	M	=	0.000
				N	=	0.000
ABSOLUTE-VALUES	A	=	0.000	B	=	0.000
				C	=	0.000
	D	=	0.000	E	=	0.000
				F	=	0.000
PROJECTION-PLANE	X	=	0.000	Y	=	0.000
				Z	=	0.000
	U	=	0.000	V	=	0.000
				W	=	1.000
RAD.-CORR. DIREC.	U	=	1.000	V	=	0.000
				W	=	0.000
ADD. PARAMETER	1	=	0.000	2	=	0.000
				3	=	0.000
	4	=	0.000	5	=	0.000
				6	=	0.000
ITERATION START VALUE [ITERATION CONTROL]						
X	=	0.000	[]	Y	=	0.000 []
				Z	=	0.000 []
U	=	1.000	[Y]	V	=	0.000 [Y]
				W	=	0.000 [C]
A	=	0.000	[]	B	=	0.000 []
				C	=	0.000 []

Eistste anche un modo molto piu' guidato che consiste nell'utilizzo del comando SETITR (settare valori di iterazioni) e che ci permette di compilare i campi indicati sopra tramite una comoda tabella.

```

+-----+
| SET ITERATION PARAMETERS |
+-----+

Command                               =

Element name                          (NAM) =
Element type                          (TYP) =
X Coordinate                          (X ) =
Y Coordinate                          (Y ) =
Z Coordinate                          (Z ) =
A Value                               (A ) =
Axis direction      (+- X,Y,Z) (DIR) =
Direction cosinus in X                (DRX) =
Direction cosinus in Y                (DRY) =
Direction cosinus in Z                (DRZ) =
Int/Ext characteristic  ([0]/I) (IND) =
Fix X Coordinate ?                   ([N]/Y) (FIX) =
Fix Y Coordinate ?                   ([N]/Y) (FIY) =
Fix Z Coordinate ?                   ([N]/Y) (FIZ) =
Fix direction ?                      ([N]/Y) (FID) =
Fix A Value ?                        ([N]/Y) (FIA) =

```

A questo punto si puo' richiamare BLDCYL che deve funzionare corretamente. Si vede pero che questo approccio è un po pesante e poco sicuro per quanto se uno si sbaglia su cosa e dove mettere, i risultati possono anche essere sbagliati. Si consiglia a questo punto di usare un "trucco". In effetti cercheremo di sfruttare il fatto che la genrezione è un comando che compila corretamente la parte dei valori iniziali d'iterazione. Per il nostro esempio questo significa di utilizzare un comando GENCYL indicando il nome del elemento CIL (dove abbiamo già ragrupato tutti i punti). Non importa quanti punti e su che altezza si generano (non non intendiamo di tastare questi punti) – importante è solo di indicare la sua posizione X,Y,Z molto approssimativa e la sua direzione. Ecco come si presenta il NOM del elemento CIL dopo l'esecuzione del GENCYL:

ELEMENT DATA					
ELEMENT	-TYPE	=	CYL	COORD.-SYSTEM	= CSY (0)
				EVAL.-SYSTEM	= (0)
PROJECTION-TYPE	=			INT. / EXT.	= 0
RAD.-CORR. TYPE	=			RAD.-CORR. VALUE	=
EVAL. -TYPE	=			SIGMA-DISPLACEM.	= 0.000
CALCUL. -TYPE	=			CODE	=
COORDINATES	X =	0.000	Y =	0.000	Z = 0.000
SPACE-ALIIINMENT	U =	0.000	V =	0.000	W = 1.000
PLANE-ALIGNMENT	L =	1.000	M =	0.000	N = 0.000
ABSOLUTE-VALUES	A =	20.000	B =	0.000	C = 0.000
	D =	0.000	E =	0.000	F = 0.000
PROJECTION-PLANE	X =	0.000	Y =	0.000	Z = 0.000
	U =	0.000	V =	0.000	W = 1.000
RAD.-CORR. DIREC.	U =	1.000	V =	0.000	W = 0.000
ADD. PARAMETER	1 =	0.000	2 =	0.000	3 = 0.000
	4 =	0.000	5 =	0.000	6 = 0.000
ITERATION START VALUE [ITERATION CONTROL]					
X =	0.000	[Y]	Y =	0.000	[Y]
U =	1.000	[Y]	V =	0.000	[Y]
A =	20.000	[D]	B =	0.000	[]
			Z =	0.000	[Y]
			W =	0.000	[C]
			C =	0.000	[]

Adesso possiamo eseguire BLDCYL indicando il tipo del raggio compensazione “Explicit” , l’indicazione “Esterno/Interno” e poi indicando anche il diametro del tastatore usato per la coretta compensazione del raggio.

GENCYL (NAM=CIL, XCO=0, YCO=0, ZCO=0...)
BLDCYL (NAM=CIL, RAD=6, CTY=EX, INO=0)

Lo stesso procedimento si consiglia anche per i cono dove per impostare i valori iniziali del calcolo si usa la generazione del cono GENCON sequita da calcolo del cono BLDCON.

Riassunto Procedura Calcolo CILINDRO / CONO da punti

- Misura dei punti sulla suuperficie del elemento (cono o cilindro) usando cerchi, assi, in modalita punti singoli o scansione:

MECIR ...
MECIR...
MEAXI...
MEAXI...

- Raggruppamento dei punti con il comando COLAPT

COLAPT ...

- Impostare i valori approssimativi per il calcolo utilizzando un comando di generazione

(GENCYL per cilindro, GENCON per cono, oppure tramite SETITR)

GENCYL ... (oppure GENCON...oppure SETITR)

- Eseguire il calcolo del elemento (BLDCYL per il cilindro, BLDCON per il cono). Impostare i parametri del raggio compensazione in questo comando.

BLDCYL... (oppure BLDCON...)

Questo approccio permette di costruire da una qualsiasi sequenza dei punti un cilindro o un cono. Alcune volte pero’ si può risparmiare tempo utilizzando un altro comando per calcolo del cilindro e cioè RND_CYL – per usare questo comando si devono misurare solo cerchi su diversi sezioni e poi raggruppare questi cerchi con il comando RND_CYL. (E’ chiaro che in questo caso la direzione iniziale del calcolo puo’ essere calcolata dal comando RND_CYL prendendo l’asse che passa tra i centri di tutti i cerchi raggruppati).

Esempio:

- Misura cerchi

MECIR (NAM=C(1))

MECIR (NAM=C(2))

MECIR (NAM=C(3))

- Calcolo cilindro

RND_CYL (SRC=(C(1),C(2),C(3)), DST=CIL)