Coordonate de puncte de interes tehnic, calculate în AutoCad

Cercetător ştiinţific gr.I, Virgiliu Streian
Universitatea "Tibiscus" , Timişoara

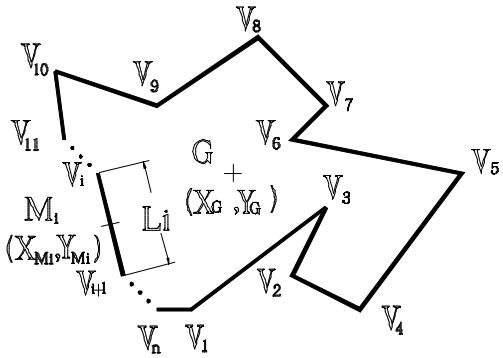
ABSTRACT. The gravity center of a polygonal closed surface is one of the most frequently used technical applications necessary for a computer aided design project. The algorithm for calculating this gravity center and its implementation in AutoLisp of AutoCad are displayed below. Concerning the programming technique, this AutoLisp program is a perfect example of AutoCAD's CAD applications development.

1 Consideraţii teoretice asupra calculării centrului de greutate al unui contur poligonal închis

Conform teoriei matematice, coordonatele XG , YG ale centrului de greutate G al unui contur poligonal închis se calculează după formulele:

$$\mathbf{X_G} = \frac{\sum_{i=1}^{n}(Xmi * Li)}{P}$$

$$\mathbf{Y_G} = \frac{\sum_{i=1}^{n}(Ymi * Li)}{P}$$



unde :

$P = \sum_{i=1}^{n} Li$ reprezintă perimetrul conturului poligonal

$Li = \sqrt{(X_i - X_{i+1})^2 + (Y_i - Y_{i+1})^2}$ reprezintă lungimea laturii i .

Mi = mijlocul laturii Li

Xmi , Ymi = coordonatele punctului Mi

Vi = vârful i al poligonului (vertex)

2 Cerințe minimale de realizare a unei aplicații de calcul al centrului de greutate al unui contur poligonal închis în Autocad

Aplicația AutoLISP, prin care se calculează coordonatele Xg , Yg ale centrului de greutate G, consideră conturul poligonal închis realizat în următoarele condiții:

1. este creat cu comanda AutoCAD , PLINE;
2. este un contur închis obținut prin utilizarea opțiunii Close a comenzii PLINE;
3. conține atât segmente de linie cât şi arce de cerc.

În aceste condiții aplicația trebuie să îndeplinească următoarele funcții:

1. Selectare: selectează o singură entitate grafică ;
2. Verificare: verifică că entitatea grafică selectată este o polilinie şi dacă da se verifică că polilinia reprezintă un contur închis;
3. Preluare coordonate (creare listă) vertexuri polilinie: în cazul vertexurilor segment de dreaptă se preiau coordonatele vertexului ca atare, iar în cazul vertexurilor arc de cerc se calculează coordonate de puncte intermediare;
4. Calculare : calculează coordonatele centrului de greutate al conturului trasat;
5. Afişare : desenează o cruciuliță în centrul de greutate al conturului şi afişează coordonatele centrului de greutate .

Lansare în execuție a aplicației se face pe prompterul Command: al mediului AutoCad după cum urmează:

Command: (load "cengre.lsp>")↵
Command: cengre↵

Selectați un contur închis ...
Select objects:<PICK cu mouse-ul conturul închis>
XG=345.45 YG=456.78

Command:

unde :
cengre.lsp - este specificatorul unui fişier .lsp ce conține funcțiile AutoLISP ale aplicației.
cengre - este numele funcției AutoLISP ce startează aplicația; funcția este definită în cadrul fişierului cengre.lsp ca şi o comandă AutoCAD printr-o linie de forma: (defun C:CENGRE ()).

3 Tehnici de programare utilizate pentru realizarea aplicației de calcul al centrului de greutate al unui contur poligonal închis

Entitatea grafică "polilinie" este o entitate compusă. Ea se prezintă utilizatorului ca o succesiune de înregistrări de tipul :
entitate principală (start polilinie)
subentitate 1 (vertex)
subentitate 2 (vertex)
subentitate 3 (vertex)
subentitate 4 (vertex)
…etc…
subentitate n (vertex)
stop entitate principală

Concret, pentru o polilinie, structura parțială a listei asociată este următoarea :

```
(          ; entitate principală (start polilinie)
  ( -1 . <Entity name:nnnnnnnn> )
  ( 0 . "POLYLINE")
```

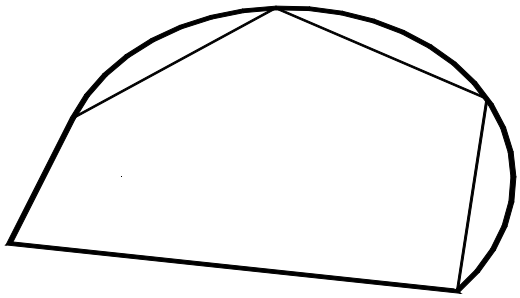
```
   ( 66 . 1)
   ( 70 . 1)      ; 1 = polilinie închisă / 0 = polilinie deschisă
  ( 10  0.0  0.0  0.0)
)

(       ;  subentitate 1
   ( -1 . <Entity name:nnnnnnnn> )
   ( 0 . "VERTEX")
....( 42 . 0.0)        ; vertex segment de dreaptă
   ( 10  1.0  1.0  0.0) ; coordonate vertex 1 - segment de dreaptă
)
(           ; subentitate 2
   ( -1 . <Entity name:nnnnnnnn> )
   ( 0 . "VERTEX")
....( 42 . -0.5345)     ; vertex arc de cerc
   ( 10  5.0  5.0  0.0) ; coordonate început vertex 2 - arc de cerc
)

(           ; sfârşit subentităţi polilinie("SEQEND")
   ( -1 . <Entity name:nnnnnnnn> )
   ( 0 . "SEQEND")
)
```

În entitatea principală perechea cu punct: (70 . x) indică faptul că polilinia este închisă (x = 1) sau deschisă (x = 0), iar în subentităţile de tip vertex lista: (10 x y z) conţine coordonatele vertexului respectiv (nod polilinie), iar perechea cu punct: (42 . x) indică, funcţie de x, faptul că vertexul startează un segment de dreaptă (x = 0.0) sau un arc de cerc (x <> 0.0).

Aplicaţia va trebui să citească fiecare vertex al poliliniei; vertexul poate să reprezinte un segment de dreaptă sau un arc de cerc. Se extrag coordonatele vertexului (cod DXF 10), iar pentru cazul în care vertexul reprezintă un arc de cerc (cod DXF 42) se împarte arcul de cerc în segmente de arc foarte mici, calculându-se şi preluându-se apoi coordonatele acestor segmente de arc:



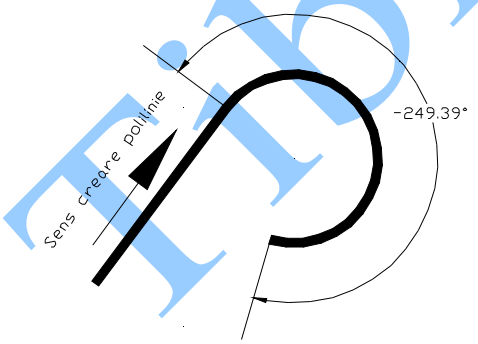
Punctele rezultate în urma procesării înregistrărilor de tip vertex vor constitui o listă de forma:

((x1 y1) (x2 y2) (x3 y3) ...)

care vor intra în modulul de calcul al centrului de greutate.

Şi acum să analizăm puţin perechea cu punct (42 . x), x fiind un număr real, din lista asociată unui vertex. Această pereche va conţine în partea dreaptă 0.0 pentru orice segment de dreapta al poliliniei şi ceva diferit de 0.0, pentru un arc de cerc. În cel din urmă caz x reprezintă tangenta sfertului de unghi la centru al arcului considerat (unghiul la centru este cuprins între 0.0 şi 2π).

De exemplu în cazul poliliniei :



creată după următoarea secventă de comenzi AutoCAD:

Command: PLINETYPE
Enter new value for PLINETYPE <2>: 0↵
Command: PLINE ↵
From point: 2,2↵
Current line-width is 0.00

Arc/Close/Halfwidth/Length/Undo/Width/<Endpoint of line>: w↵
Starting width <0.00>: 0.2↵
Ending width <0.20>: ↵
Arc/Close/Halfwidth/Length/Undo/Width/<Endpoint of line>: 5,6↵
Arc/Close/Halfwidth/Length/Undo/Width/<Endpoint of line>: a↵
Angle/CEnter/CLose/Direction/Halfwidth/Line/Radius/Second pt/Undo/Width/<Endpoint of arc>: 6,3↵
Angle/CEnter/CLose/Direction/Halfwidth/Line/Radius/Second pt/Undo/Width/<Endpoint of arc>: ↵
Command:

NOTĂ:

PLINETYPE controlează atât crearea unei noi polilinii prin comanda PLINE cât şi conversia unei polilinii existente creată într-o versiune anterioara de AutoCad.

0 Poliliniile din desene vechi nu sunt convertite atunci când deschidem un nou desen, PLINE crează un format vechi de polilinie.

1 Poliliniile din desene vechi nu sunt convertite atunci când deschidem un nou desen; PLINE crează polilinii optimizate.

2 Poliliniile din desene vechi sunt convertite atunci când deschidem un nou desen; PLINE crează polilinii optimizate

Vertex-ul de unde începe arcul de cerc al poliliniei are următoarea lista de subentitate asociată:

((-1 . <Entity name: 60000066>) (0 . "VERTEX") (8 . "0") (10 5.0 6.0 0.0) (40 . 0.0) (41 . 0.0)
(42 . -1.90857) (70 . 0) (50 . 0.0) (71 . 0) (72 . 0) (73 . 0) (74 . 0))

Perechea cu punct : (42. -1.90857), identifică tangenta sfertului de unghi la centru al arcului considerat .

arctan(-1.90857) = - 62.3476 °

- 62.3476 x 4.0 = - 249.39 ° (vezi desenul de mai sus)

semnul "-" semnifică faptul că arcul este creat în sens invers trigonometric

Arcul de cerc al poliliniei va fi deci definit prin :

Start point :5.0 , 6.0 (coordonate vertex în lista: (10 5.0 6.0 0.0))

Angle :-249.39° (rezultă din perechea: (42 . -1.90857))

End point :6.0 , 3.0 (în vertex-ul următor)

Funcția AT de mai jos determină şi calculează în grade sexazecimale unghiul asociat codului DXF 42 înmulțit cu 4:

```
(defun C:AT ()
 (setq NEL      ; preia lista asociată unui vertex de polilinie - arc de cerc -
 (entget
(nth 0 (nentsel))
)
 )
 (setq TG (cdr (assoc 42 NEL)))        ; extrage din ea tangenta unui unghi
 (setq ATGr (atan TG))          ; determină unghiul
 (setq ATG
 (*
   (cvunit ATGR "radian" "degree")   ;conversie unghi din radiani în grade sexazecimale
   4
 )
 )
)
```

În principiu, pentru identificarea entității principale (polilinie) şi a vertexurilor aferente (subentități), precum şi a procesării vertexurilor, se vor utiliza următoarele funcții AutoLISP (acces la entități grafice şi procesare coordonate), preyentate în tabelul următor.

Funcția ENTSEL pretinde selectarea unui singur obiect şi returnează o listă de forma (<Entity name:nnnnnnnn.(x y x)) ce conține numele entității principale şi coordodatele punctului în care s-a făcut selecția respectivei entități; dacă nu s-a selectat nimic atunci răspunsul funcției est nil.

Funcția ENTNEXT returnează numele de entitate pentru prima entitate neştearsă din baza de date grafică ce urmează lui <ename> .

ENTNEXT returnează atât nume de entități pricipale cât şi nume de subentități.

Funcție	Operație executată
(ensel [<prompt>] >)	Returnează o listă ce contine numele entității principale şi coordodatele punctului în care s-a făcut selecția
(entnext <ename>)	Returnează numele de entitate din baza de date grafică ce urmează numelui indicat prin ename
(entget <ename>)	Returnează lista asociată entității indicate
(angle <p1> <p2>)	Returnează unghiul în radiani al dreptei definite de punctele argument: p1 p2
(distance <p1> <p2>)	Returnează distanța 3D între două puncte
(polar <p> <angle> <distance>)	Returnează un punct 3D UCS aflat la un unghi şi la o distanță dată față de alt punct

Funcția ENTGET extrage entitatea specificată prin numele de entitate <ename> din baza de date grafică şi o returnează ca o listă asociată entității respective; funcția assoc permite apoi extragerea de date dintr-o astfel de listă ; în caz de insucces funcția răspunde cu nil. Obiectele într-o astfel de listă sunt codificate în conformitate cu codul de grupa DXF AutoCAD aplicat fiecărei parți de date ce caracterizează entitatea (perechi cu punct de forma codi . vali) . Prin acestă funcție se poate avea acces atât la entitatea principală cât şi la componente ei (entități de tip vertex în cazul poliliniei). Codul DXF 0 va puncta inițial startul poliliniei (“PLINE”) , ulterior vertexurile poliliniei (“VERTEX”), iar în final sfârşitul poliliniei (“SEQEND”)

Structura generală a unei liste asociată unei entități este următoarea :

```
(list
(cons -1  <Entity name:nnnnnnnn> )
(cons 0  <tip_entitate> )
(cons <cod3> <val3> )
     .
 (cons <codn> <valn> )
)
```

Pentru o polilinie, structura parțială a listei asociată este următoarea :

```
 (list          ; entitate principală (start polilinie)
```

```
  (cons -1  <Entity name:nnnnnnnn> )
  (cons 0 "POLYLINE")
  (cons 66 1)
  (cons 70 1)      ; 1 = polilinie închisă / 0 = polilinie deschisă
 (list 10 0.0 0.0 0.0)
)
(list          ;  subentitate 1
  (cons -1  <Entity name:nnnnnnnn> )
  (cons 0 "VERTEX")
....(cons 42  0.0)       ; vertex segment de dreaptă
  (list 10 1.0 1.0 0.0)   ; coordonate vertex 1 - segment de dreaptă
)
(list          ; subentitate 2
  (cons -1  <Entity name:nnnnnnnn> )
  (cons 0 "VERTEX")
....(cons 42  -0.5345)     ; vertex arc de cerc
  (list 10 5.0 5.0 0.0)   ; coordonate început vertex 2 - arc de cerc
)
(list          ; sfârşit subentităţi polilinie("SEQEND")
  (cons -1  <Entity name:nnnnnnnn> )
  (cons 0 "SEQEND")
)
```

Funcţia ANGLE returnează unghiul în radiani a unei drepte imaginare ce uneşte două puncte <p1> şi

<p2>, din UCS-ul curent; dacă se dau puncte 3D, dreapta este proiectată pe planul de bază XOY curent. De exemplu:

(angle (list 1 1) (list 1 4)) ;returnează 1.5708 adică π/2

Funcţia DISTANCE returnează distanţă 3D între două puncte. De exemplu:

(distance (list 1 1) (list 1 4)) ;returnează 3

Funcţia POLAR returnează un punct 3D UCS aflat la un unghi şi la o distanţă dată faţă de un punct P indicat. De exemplu:

(polar (list 1 1) (/ pi 2) 3) ;returnează (1 4)

Exemplu de program ce livrează lista entităţii principale şi a subentităţiilor unei polilinii:

```
(defun C:LV ()
;;
;;   Afişează listele asociate  ale unei entităţi compuse de tip polilinie
```

```
;;
;; autor : ing. V.STREIAN
;;
(setq EN (nth 0 (entsel)))     ; selecteaza entitate si nume entitate in EN
(setq EL (entget EN))      ; lista entitate in EL
;;
;; următoarea linie se va înlocui cu o secvență de program ce testază dacă entitatea selectată
;; este o polilinie şi dacă polilinia este închisă.
;;
(princ EL)              ; afiseaza lista asociata entitate (principala)
;;
;;
(while EN               ; cât timp avem EN executa ....
 (setq ELV (entget (entnext EN))) ; lista asociata vertex in ELV
 ;;
 (if                    ; am terminat toate subentitatiile poliliniei ?
  (=
   (cdr
    (assoc 0 ELV)
   )
   "SEQEND"
  )
  ;;
  (progn    ; THEN -->da
   (setq EN nil)  ;      .... forteaza iesirea din while
  )
  ;;
  (progn    ; ELSE -->nu (procesează vertex)
   (terpri)             ; salt de linie
   (princ ELV)              ; afiseaza lista asociata subentitatii (vertex)
  ;;
  ;;
  (setq EN (cdr(assoc -1 ELV)))  ; nume entitate utilizat în determinarea  entității urmatoare
  )
 )
)
```

4 Sursa aplicației AutoLisp care calculează coordonatele centrului de greutate al unui contur poligonal închis

```
;
   ;*******  CENGRE.LSP *******
   ; Calculeaza coordonatele centrului de greutate al unui contur inchis
   definit
   ; ca entitate "POLYLINE"
   ;;
   ;; autor : ing. V. Streian
   ;;
   ; NOTA :
   ;    1.  conturul este creat cu comanda AutoCAD  PLINE precedata de
   comada PLINETYPE...0;
   ;    2.  conturul este un contur inchis obtinut prin utilizarea optiunii
   Close a comenzii PLINE;
   ;    3.  contine atat segmente de linie cat si arce de cerc; doua arce nu
   au voie sa fie  unul dupa altul.
   ;
   ;;******* MAIN FUNCTION *******
   ;;              Functia CENGRE
   ;
   ;  calculul centrului de greutate al unui contur poliliniar inchis
   ;
   ;
   ;
   (defun C:CENGRE ()
   (LISTA_VERTEX)      ; determina vertex-urile -- LVERTEX
    (if (= LVERTEX nil)
      (progn ; THEN -- > A fost detectata o eroare
        (princ)
      )
      (progn ; ELSE -- > Calculez centrul de greutate
       ;;  Se extrag pe rand doua cite doua coordonatele varfurilor
   consecutive
       (setq I 0)
       (setq XMI 0)  ; X mijloc latura I
       (setq YMI 0)  ; Y mijloc latura I
       (setq LI  0)  ; lungime latura I
       (setq XG 0)   ;; X centru de greutate
```

```
    (setq YG 0)   ;; Y centru de greutate
    (while (< I (- (length LVERTEX) 1))
     ;
     ;; determin laturi si puncte de mijloc
     ;
     (setq LI (distance (nth I LVERTEX) (nth (+ i 1) LVERTEX)))
     (setq XI (nth 0 (nth I LVERTEX)))
     (setq YI (nth 1 (nth I LVERTEX)))
     (setq XIp1 (nth 0 (nth (+ I 1)  LVERTEX)))
     (setq YIp1 (nth 1 (nth (+ I 1)  LVERTEX)))
     (setq XMI (* 0.5 (+ XIp1 XI)))
     (setq YMI (* 0.5 (+ YIp1 YI)))
     ;
     ;; determin centrul de greutate
     ;
     (setq XG (+ XG (/ (* XMI LI) PERIMETRU)))   ;; X centru de
greutate
     (setq YG (+ YG (/ (* YMI LI) PERIMETRU)))   ;; Y centru de
greutate
     (setq I (+ I 1))
    ) ; end while
   )  ; end prog
 ) ; end if

; Afisez pe display coord. centrului de greutate.
 (print "XG = ")  (prin1 XG)
 (print "YG = ")  (prin1 YG)
 (princ)
 )      ; end functie
;
 (defun LISTA_VERTEX ()
;; determina vertex-urile poliliniei
;;  arcele de cerc sunt discretizate
;;  (inlocuite cu segmente foarte mici)
;; iesire:
;;  LVERTEX - lista vertexurilor
 (setq LVERTEX nil)            ; initializare lista vertex-uri
 (princ "Selectati un contur închis ...")
 (setq EN (nth 0 (setq SS (entsel))))     ; selecteaza entitate si nume
entitate in EN
```

```
 (setq EL (entget EN))          ; lista entitate in EL
 (princ EL)                     ; afiseaza lista asociata entitate
(principala)
 (if
  (and
   ( = "POLYLINE" (cdr (assoc '0 EL))) ; este o Polilinie ce am
selectat?
   ( = 1 (cdr (assoc '70 EL))) ; este ea inchisa cu "Close" ?
  )
  (progn    ; Then --> citesc vertexuri
   (command "AREA" "Object" ss "")      ; raspunde cu perimetrul
conturului
   (setq PERIMETRU (getvar "perimeter")) ;
   (while EN                ; cat timp avem EN executa ....
    (setq ELV (entget (entnext EN)))   ; lista asociata vertex in ELV
    (if                     ; am terminat toate subentitatiile poliliniei ?
     (=
      (cdr
       (assoc 0 ELV)
      )
      "SEQEND"
     )
     (progn                 ; THEN -->da
      (princ ELV)
      (setq EN nil)         ;      .... forteaza iesirea din while
     )
     (progn                 ; ELSE -->nu
      (setq PV (cdr(assoc 10 ELV))) ; coordonate vertex
      (setq LVERTEX (append LVERTEX (list PV))) ; adauga in lista
un vertex
      (if                   ; test daca vertexul este inceputul unui arc
       (/=
        (setq UNGHI_D       ; in radiani
         (*
         4
         (atan
          (cdr
           (assoc 42 ELV)
               )
          )
```

```
        )
       )
       0.0
      )
      (progn    ; then tratare arc
       ;; urmeaza discretizarea arcului de cerc
       (setq ENC  (cdr(assoc -1 ELV))) ; nume entitate curenta
       (setq CENTRU (osnap PV "_cen"))   ; coordonate centru arc --
PV vertex inceput arc
       (setq RAZA (distance PV CENTRU))  ; raza arcului
       (setq UNGHI_I (angle CENTRU PV))    ; unghi inclinatie raza
(radiani)

       ; determinare sens parcurgere
       (if (> UNGHI_D 0)              ;  in radiani:
        (setq UNGHI_R 0.01)       ; ratia de calcul coordonate discrete
        (setq UNGHI_R -0.01)      ; ratia de calcul coordonate discrete
       )
       ;;
       (setq NR_ITER (+ 1 (FIX (/ UNGHI_D UNGHI_R))); numar de
iteratii
       ;;
       ; calcul coordonate discrete arc
       ;;
       (setq UNG UNGHI_I)
       (while (>(setq NR_ITER (- NR_ITER 1)) 0)
        (progn
         (setq UNG (+ UNG UNGHI_R))     ; unghi inclinatie dreapta -
in radiani
         (setq PV  (polar CENTRU UNG RAZA)) ; punct discret arc
         (setq LVERTEX (append LVERTEX (list PV))) ; adauga in
lista un vertex
        ) ; end prog
       ) ; end while
      ) ; end prog
     )  ; end if
    (setq EN (cdr(assoc -1 ELV)))   ; se ia entitatea urmatoare
   ) ; end if
  )  ; end prog
 )   ; end while
```

```
  ;; primul punct devine si ultimul in LVERTEX
  (setq LVERTEX (append LVERTEX (list (nth 0 LVERTEX))))
 )  ; end prog
 (progn  ; Else --> eroare: selectia nu e polilinie sau polilinia nu e
inchisa
    (*error* "Selectia nu e polilinie sau polilinia nu e inchisa !!!!")
 )
 )  ; end if polilinie si polilinie inchisa

) ; end functie
```

Bibliografie

[Dog88] **Dogaru, Dorian** - *Metode noi în proiectare, Elemente de grafică tridimensională*, Editura ştiinţifică şi enciclopedică, Bucureşti, 1988.

[HBP02] **Harrington, Burchard, Pitzer** – *AutoCAD 2002*, Editura Teora, Bucureşti, 2002.

[HCFxx] **Hello CadFANS** – *revistă românească de proiectare asistată*.

[Sim03] **Simion, Ionel** – *AutoCAD 2000 – Aplicaţii*, Editura Teora, Bucureşti, 2003.

[Sta93] **Stăncescu, Constantin** – *AutoCAD, manual de iniţiere*, Editura FAST 2000, Bucureşti, 1993

[Sta96] **Stăncescu, Constantin** – *AutoLISP, manual de programare*, Editura FAST 2000, Bucureşti, 1996.