UDK: 519.6(047)=861
COSATI: 12-01, 09-02

O jednom postupku za utvrđivanje pripadnosti tačke poligonalnoj oblasti

Dr Zoran Drašković, dipl.meh.[1)]

Predložen je još jedan postupak za utvrđivanje pripadnosti tačke zatvorenoj poligonalnoj oblasti. Njegovo kratko vreme rada pokazano je na primeru hipotetičkog poligona sa hiljadu temena.

Ključne reči: **Poligon, prost poligon, pripadnost tačke poligonu.**

U v o d

U okviru rada na programskom paketu *MOBA* (koji obezbeđuje kreiranje i pretraživanje skupova motorskih podataka na eksternim memorijskim jedinicama, v. [1]), ukazala se potreba da se u taj paket implementira postupak logičke kontrole ulaznih podataka u odnosu na tzv. anvelopu aviona i/ili motora. Budući da se anvelopa aviona ili motora može opisati nizom uzastopno povezanih duži u ravni, taj problem kontrole se, u suštini, sveo na utvrđivanje ležanja neke tačke u unutrašnjoj oblasti ili na samoj konturi poligona obrazovanog od tih duži (segmenata). Tom prilikom su razmotrena dva, iz literature poznata, algoritma za utvrđivanje te pripadnosti tačke poligonalnoj oblasti, ali je predložen i jedan, čini se nov, postupak i upravo o njemu će ovde biti najviše reči.

Definicija poligona

Pod *poligonom* (ili zatvorenom poligonalnom linijom) podrazumeva se unija duži A_1A_2, A_2A_3, ... , A_iA_{i+1}, ..., $A_{n-1}A_n$, A_nA_1 čije krajnje tačke (temena poligona) leže u istoj ravni. Pretpostavlja se da ni jedan par duži (stranica poligona), ne računajući krajnje tačke* (temena), nema zajedničkih tačaka − tada se kaže da je poligon *prost* (ili bez samopresecanja); u protivnom, poligon je *složen* i ima bar jedan samopresek.

Prosti poligoni se mogu podeliti na *konveksne* i *nekonveksne*: kaže se da je poligon konveksan (ispupčen) ako su svi njegovi unutrašnji uglovi udubljeni − ako to nije slučaj, poligon je nekonveksan. U daljem izlaganju biće reči o prostim, u opštem slučaju nekonveksnim, poligonima.

Utvrđivanje pripadnosti tačke poligonalnoj oblasti

Kao kriterijum pripadnosti neke tačke zadatoj poligonalnoj oblasti u literaturi se najčešće sreće onaj naveden na str. 108 u [2] − to je *neparnost*** broja preseka poluprave iz te tačke i poligona. Međutim, činjenica da se radi o pripadnosti samo *unutrašnjoj* oblasti poligona (jer se posebno utvrđuje pripadnost zadate tačke poligonalnoj konturi!), kao i činjenica da bi trebalo ispitati odnos *svih* stranica poligona i uočene poluprave (što izgleda relativno zametan posao u slučaju velikog broja temena), naveli su na pomisao da bi bilo uputnije da se uoči neko drugo svojstvo (umesto neparnosti pomenutog broja preseka) koje bi utvrđivalo postojanje pripadnosti zadate tačke *zatvorenoj* poligonalnoj oblasti (tj. i unutrašnjoj oblasti poligona i njegovoj konturi), a pri tom bilo *nezavisno* u odnosu na sve moguće situacije.

Treba priznati da su, zbog jakog utiska koji na čitaoca ostavlja u [4] (str. 166) izneti stav da svaki algoritam treba zasnivati na nekom svojstvu *invarijantnosti* u proučavanom procesu, prethodna razmišljanja bila motivisana zapravo pokušajem da se formuliše *invarijantan* kriterijum pripadnosti poligonalnoj oblasti.

U takvim nastojanjima naišlo se, u publikaciji [5] (str. 68-69 i 184-188), na još jednu ideju za rešavanje problema pripadnosti neke tačke unutrašnjoj oblasti ili konturi prostog poligona***: neka tačka se nalazi u poligonu ili izvan poligona, zavisno od toga da li je *zbir svih uglova* pod kojima se iz te tačke vide pojedinačne stranice poligona jednak 2π ili je 0 .

Međutim, originalnost iznete ideje i jednostavnost samog postupka za utvrđivanje pripadnosti date tačke zadatoj poligonalnoj oblasti nisu bili razlog da ne budemo oprezni. Naime, naišavši i na odgovarajući program u kojem figuriše sistemska funkcija *ATAN2* (algoritam je, u publikaciji [5] iz koje je uzet, bio propraćen i programskim kodom!), setili smo se da je pre više godina, u toku rada na paketu programa *MORINO* (v. [6]), bilo izvesnih poteškoća pri korišćenju te funkcije****. Da bismo otklonili nedoumicu oko mogućnosti primene funkcije *ATAN2*, smatrali smo da

1 Vojnotehnički institut VJ, 11000 Beograd, Katanićeva 15

* Koje se, po pretpostavci, sve međusobno razlikuju!

** Što se koristi, na primer, za potrebe razrešavanja zadatka senčenja poligonalnih površina (v. [3], str.83-85).

*** Okolnost da ta ideja potiče od saradnika (Žan-Kloda Ribea) Nacionalnog centra za naučna istraživanja (CNRS), vrlo ugledne francuske ustanove, bila je razlog više da se i taj postupak uzme u obzir.

****Suština je bila u tome da je takva funkcija višeznačna, a ako se posmatra samo jedna njena grana, ta funkcija ima *prekid* – taj problem je prezaviđen razvijanjem i implementiranjem u paket *MORINO* posebnog funkcijskog potprograma *FATAN* (v. [6]).

bi bilo uputno zadržati se na nekim naročitim slučajevima, a to su zapravo situacije kada je zadata tačka na samoj poligonalnoj konturi. I zaista, bojazni su se obistinile, jer kad smo uzeli dve tačke koje leže na međusobno paralelnim, ali (s obzirom na način "obilaska" konture) suprotno orijentisanim segmentima poligonalne konture, odgovarajući program dao je odgovor da jedna tačka pripada poligonalnoj oblasti, a druga da ne pripada!

Takav rezultat bio je razlog da se (budući da je odlučeno da se u programskom paketu *MOBA* usvoji prethodni postupak, i to pre svega zbog izuzetne kratkoće programa koji mu odgovara*, ali i okolnosti da u problemima na koje će se primenjivati taj algoritam neće biti više od nekoliko desetina tačaka anvelope, tj. temena odnosne poligonalne oblasti**), modifikuje odgovarajući program kako bi se otklonila mogućnost da iz razmatranja bude isključena neka tačka poligonalne konture prilikom logičke kontrole podataka u odnosu na anvelopu. Međutim, ovde se nećemo baviti tim utanačenjima, nego će čitaočevoj pažnji biti preporučen

Jedan novi postupak za utvrđivanje pripadnosti tačke zatvorenoj poligonalnoj oblasti

Naime, usvajanjem prethodnog algoritma u paketu *MOBA* nisu prestala naša nastojanja da se dođe do još nekog kriterijuma pripadnosti poligonalnoj oblasti, koji bi bio nezavisan u odnosu na sve moguće situacije. I zaista, iz postupka analize *usmerenosti* duži *najbližih* zadatoj tački, uspeli smo da formulišemo još jedan uslov pripadnosti zatvorenoj poligonalnoj oblasti i da pri tome obuhvatimo i specijalne slučajeve. O tome će biti reči u ovom odeljku, pri čemu poligon smatramo unijom *poluzatvorenih* intervala $[A_i, A_{i+1})$.

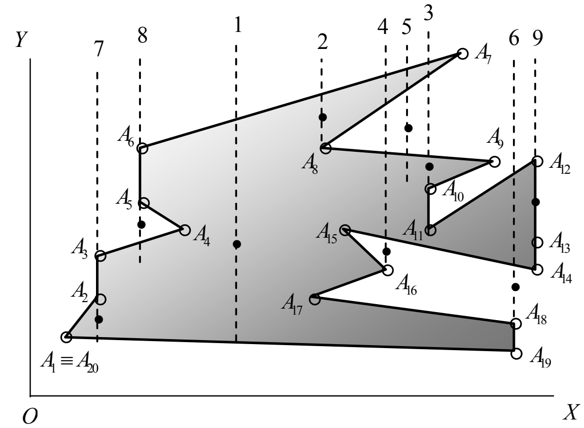
Kriterijum pripadnosti poligonalnoj oblasti. Ako sve stranice poligona, već prema tome da li je*** $x_i < x_{i+1}$, $x_i = x_{i+1}$ ili $x_i > x_{i+1}$, klasifikujemo kao duži *klase* +1, 0 ili -1 respektivno, onda se može formirati sledeća šema mogućih položaja zadate tačke i njoj, sa gornje i donje strane, "najbližih" duži:

Slučajevi na slici 1:	1	2	3	4	5	6	7	8	9
klasa "gornje" duži:	+	+	+	-	-	-	0	0	0
zadata tačka:	•	•	•	•	•	•	•	•	•
klasa "donje" duži:	-	+	0	-	+	0	-	+	0
pripadnost poligonu:	$\in$	$\in$	$\in$	$\notin$	$\notin$	$\notin$	$\in$	$\notin$	$\in$

Šema 1

pri čemu je, u donjem delu šeme, za svaku od tih mogućnosti označeno da li će zadata tačka pripadati ($\in$) ili ne pripadati ($\notin$) posmatranoj poligonalnoj oblasti. Zaključak o pripadnosti poligonalnoj oblasti tačke ako joj je sa gornje strane "najbliža" duž klase +1, a sa donje strane duž klase -1 (to je prvi slučaj u prethodnoj šemi), lako se izvodi ako se baci pogled na sliku 1; pri tome je od suštinskog značaja pretpostavka *o obilaženju konture* u smeru kretanja kazaljke na časovniku (što bi trebalo da bude provereno u fazi unošenja podataka o anvelopi).

Isto tako, sa te slike se odmah vidi da tačka koja je između dve duži, od kojih je gornja klase -1 , a donja klase +1 , ne pripada poligonalnoj oblasti (peti slučaj u šemi). Moglo bi se reći da su to dva "najvažnija"**** slučaja mogućih odnosa klasa najbližih stranica poligona koje preseca ordinata kroz zadatu tačku $P(x_P,y_P)$ čija se pripadnost poligonu ispituje i upravo je to ono što treba utvrditi posle pronalaženja tih "najbližih" stranica. Ipak, postoje još neke mogućnosti (preostali slučajevi u šemi) koje takođe moraju biti uzete u obzir, makar da se radi o "specijalnim" slučajevima. Radi potpunosti su na slici 1 prikazani primeri koji odgovaraju svim slučajevima u toj



šemi (broj uz ordinatu odgovara broju slučaja u šemi).

Slika 1.

Međutim, time što su uočeni ovi mogući slučajevi odnosa zadate tačke i njoj najbližih duži *između****** kojih se ona nalazi, još nisu iscrpljene sve mogućnosti pripadanja****** poligonalnoj oblasti; naime, osim što tačka može biti između dveju duži, ona može i ležati *na duži*, tj. pripadati stranici poligona, a to će biti ustanovljeno već u toku traženja duži između kojih se nalazi zadata tačka (to će se videti iz samog opisa algoritma u sledećem odeljku).

Opis algoritma. Algoritam za utvrđivanje pripadnosti zadate tačke datoj poligonalnoj oblasti otpočeo bi time što bi se za sve stranice poligona prvo utvrdilo kojoj *klasi* pripadaju, zavisno od toga da li je $x_i < x_{i+1}$, $x_i = x_{i+1}$ ili $x_i > x_{i+1}$ i "napunio" niz $KLASA_i = signum\ (\ x_{i+1} - x_i\ )$; pri tome se $KLASA_i$ odnosi na *i*-tu stranicu poligona koja je definisana sa $[A_i, A_{i+1})$ (razume se da je $A_{n+1} \equiv A_1$).

No, u istom ciklusu u kom se određuje klasa stranica poligona može da se vrši i ispitivanje da li apscisa x_P zadate tačke P pripada intervalu $[x_i, x_{i+1})$; odgovarajući deo algoritma bi bio oblika:

* Što olakšava uvid u sam program pri njegovom eventualnom kasnijem modifikovanju.

** Pa nisu od posebnog značaja prednosti nekog drugog postupka – s obzirom na manji utrošak procesorskog vremena – jer one dolaze do izražaja tek u slučaju poligona sa vrlo velikim brojem temena, o čemu će kasnije biti govora.

*** x_i, y_i su koordinate *i*-tog (i=1,...,n) temena poligona u Dekartovom (Descartes) koordinatnom sistemu *Oxy*.

**** I najopštija, jer utvrđuju pripadanje, odnosno nepripadanje poligonalnoj oblasti u svim „nedegenerisanim" slučajevima.

***** U sledećem odeljku će biti navedeno da se za vertikalnu (dakle klase 0) duž, kao njen presek sa pravom na kojoj leži, uzima njena prva tačka; stoga se na slici 1 i moglo reći da se zadata tačka između temena A_{12} i A_{13} nalazi *između* duži $[A_{12}, A_{13})$ i $[A_{13}, A_{14})$.

****** Što se tiče *nepripadanja* zadate tačke poligonalnoj oblasti, zato što se nalazi sa gornje ili donje strane poligona (a što nije obuhvaćeno šemom 1), ono se lako utvrđuje ispitivanjem parametara n^{up} i n_{down} (v. sledeći odeljak).

```
j=0
do i = 1 , n
   KLASA_i = ...
   if(KLASA_i .eq. 0) then
     if(x_P .eq. x_i) then
       j = j+1
       y_j^cross = y_i
       NIZ_j = i
     endif
  else
   if((x_P .ge. x_i .and. x_P .lt. x_i+1) .or. (x_P .gt.
     x_i+1 .and. x_P .le. x_i)) then
     j = j + 1
     y_j^cross = y_i + (y_i+1 - y_i) * (x_P - x_i) / (x_i+1 - x_i)
     NIZ_j = i
   endif
 endif
enddo
n_cross = j
```

Pri tome je izračunata i ordinata y_j^{cross} preseka prave x=x_P i odnosne stranice poligona (razume se, ako presek postoji), s tim da je u slučaju stranice poligona upravne na x-osu, ako se poklapaju ta stranica i prava x=x_P , za ordinatu preseka uzeta ordinata *prve* tačke duži $[A_i,A_{i+1})$. Takođe je uveden i pomoćni niz NIZ_j koji uspostavlja korespondenciju između rednog broja j preseka prave $x = x_P$ sa stranicama poligona i rednog broja i odnosne strane. Po završetku tog dela algoritma, promenljiva n_{cross} bi davala podatak o broju preseka prave x=x_P i stranica poligona, tj. ukupan broj stranica poligona koje se seku sa tom pravom.

Ako je n_{cross}=0, jasno je da zadata tačka P ne pripada posmatranom poligonu; u tom slučaju je ispitivanje odmah završeno.

Ako je $n_{cross} \neq 0$, tada treba odmah izdvojiti slučaj kad je n_{cross} =1 , jer to zapravo znači (s obzirom na to da je poligon definisan kao unija *poluzatvorenih* duži $[A_i,A_{i+1})$!) da prava x=x_P seče (tačnije, dodiruje) poligon u jednoj jedinoj tački i to baš u nekom njegovom temenu; tada jedino preostaje da se utvrdi da li je y_P jednako* ordinati tog temena poligona. Deo programa u kom bi se ovo ispitivalo, izgledao bi ovako:

```
if(n_cross .eq. 0) then
   type * ,'P ne pripada poligonu'
else
   if(n_cross .eq. 1) then
     if(y_P .eq. y_NIZj) then
       type * ,'P se podudara sa temenom', NIZ_j
   else
      type * ,'P ne pripada poligonu'
   endif
 else
   .
   . (*)
   .
 endif
endif
```

Ono o čemu će dalje biti reči, odnosi se na deo algoritma koji je u prethodnoj šemi označen sa *(*)* , tj. na situaciju kada je n_{cross}>1, što znači da prava x=x_P preseca *bar dve* stranice poligona. Najpre se od tih n_{cross} stranica poligona koje seče prava x=x_P pronalaze one dve koje su *najbliže*, sa gornje i sa donje strane, zadatoj tački $P(x_P,y_P)$; pri tome se uvode promenljive n^{up} i n_{down} koje će predstavljati brojeve odnosnih stranica. Posle završetka tog pretraživanja (koje eksplicitno ne navodimo!), treba utvrditi da li je možda n^{up} ili n_{down} ostalo jednako nuli, što bi značilo da zadata tačka ne pripada poligonu, jer je ili iznad ili ispod svih preseka prave x=x_P sa njegovim stranicama.

Ukoliko to nije slučaj, tj. ako je $n^{up} * n_{down} \neq 0$, prvo treba utvrditi da možda nije $n^{up} = n_{down}$, jer bi to značilo da zadata tačka leži na konturi, tj. na nekoj stranici poligona (naime, ispitivanjem da li je y_P veće ili *jednako*, odnosno manje ili *jednako* od ordinata y_j^{cros} preseka prave $x = x_P$ sa stranicama poligona omogućeno je da se odmah utvrdi eventualno podudaranje zadate tačke P sa nekim od tih preseka, tj. njeno pripadanje konturi poligona; treba naglasiti da je tom analizom obuhvaćen i slučaj kada se zadata tačka podudara sa nekim temenom konture).

Najzad, preostaje da se izvrši analiza i situacije kada je $n^{up} \neq n_{down}$, tj. kada se zadata tačka nalazi na delu ordinate *između* dve stranice poligona. Ipak, pre nego što bude razmatrana mnogostrukost mogućih kombinacija u ranije navedenoj šemi, prvo ćemo navesti kako bi izgledao deo algoritma u kom se analizira odnos parametara n^{up} i n_{down} :

```
if(n^up * n_down .eq. 0) then
  type * ,'P ne pripada poligonu'
else
  if(n^up .eq. n_down) then
    type * ,'P je na stranici', n^up ,' polgona'
  else
    .
    . (**)
    .
  endif
endif
```

Sada će biti reči o delu algoritma koji je u prethodnoj šemi označen sa (**), a odnosi se na situaciju kada je $n^{up} \neq$ $\neq n_{down}$ i zapravo je ključni deo celog postupka za utvrđivanje pripadnosti zadate tačke datoj poligonalnoj oblasti.

S obzirom na tada moguće situacije koje su prikazane na ranije navedenoj šemi (za šta je, to opet treba podvući, bitna pretpostavka da se poligon, gledano iz pozitivnog dela z-ose desno orijentisanog Dekartovog $Oxyz$-sistema, "obilazi" u smeru kretanja kazaljki na satu!), preostalo ispitivanje bi se moglo obaviti na sledeći način:

```
if((KLASAn^up * KLASAn_down .eq. 0) .and.
      ((abs(n^up-n_down) .eq. 1) .or. (abs(n^up-n_down) .
      eq. (n-1)))) then
  if(KLASAn^up .eq. 0 .and. KLASAn_down .eq. 0) then
    type * ,'P je na jednoj od susednih vertikalnih
    stranica', n_down ,' i ', n^up
  else
    if(KLASAn^up .eq. 0) type * ,'P je na vertikalnoj
     stranici', n^up
    if(KLASAn_down .eq. 0) type * ,'P je na vertikal
     noj stranici', n_down
  endif
else
  if((KLASAn^up .eq. 1) .or. ((KLASAn^up .eq. 0) .and.
   (KLASAn_down .lt. 1))) then
    type * ,'P pripada poligonu'
  else
    type * ,'P ne pripada poligonu'
  endif
endif
```

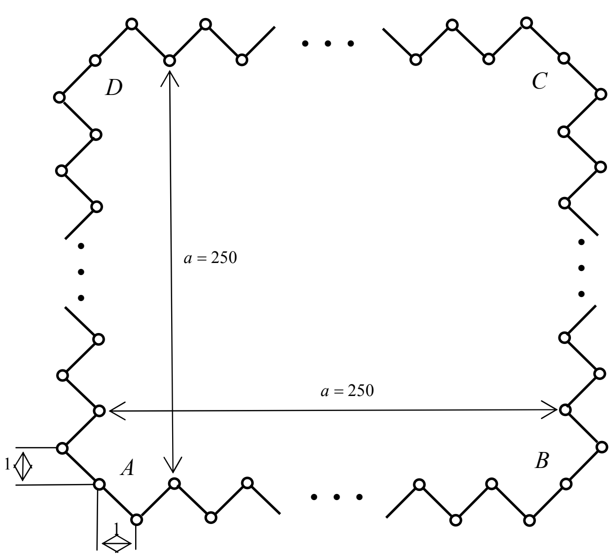
* Koordinate temena poligona, kao i zadate tačke, predstavljaju tačno zadate ulazne (a ne sračunate) vrednosti, pa stoga ima smisla takvo ispitivanje jednakosti dva realna broja!

i time bi, zapravo, u celosti bio opisan predloženi postupak za ispitivanje pripadnosti neke tačke unutrašnjoj oblasti ili konturi prostog poligona*.

Jasno je da je programski kod ranije pomenutog postupka (koji koristi funkciju *ATAN2*) neuporedivo kraći od programskog koda sada predloženog postupka. Stoga, kada je reč o dužini samog programskog koda, nije bilo dvoumljenja da se u paketu *MOBA ne usvoji* takav predlog. Međutim, o pravim razlozima zašto mu je ipak posvećeno toliko pažnje − iako on možda nije opravdao sva prvobitna očekivanja da će biti zasnovan na kriterijumu pripadnosti zatvorenoj poligonalnoj oblasti koji bi na jednostavan način obuhvatio sve moguće situacije − tek treba da bude reči.

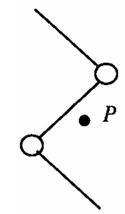
Procesorska zahtevnost predloženog postupka. Naime, uobličavanje ovog, u suštini *heurističkog* (rađenog, dakle, bez ugledanja na neke u literaturi prisutne algoritme), postupka bilo je motivisano i nastojanjem da se izbegne prekomeran broj *numeričkih* operacija i da se obavi što je moguće veći broj *upoređivanja* veličina (to je vremenski svakako manje zahtevno).

I zaista, taj algoritam je pokazao svoje prednosti u pogledu brzine rada. Primer na kom je izvršeno poređenje njegove procesorske zahtevnosti sa zahtevnošću postupka koji koristi funkciju *ATAN2* prikazan je na slici 2.



Slika 2.

Radi se o hipotetičkom primeru poligona sa 1000 (= 4 * 250) temena; tako veliki broj tačaka je zato uzet da bi se pouzdanije moglo izvršiti poređenje odnosnih procesorskih vremena (razume se, prilikom testiranja je tačka *P* zadavana i na način prikazan na slici 3, tj. između "vertikalnih zubaca", kako se ne bi neopravdano favorizovao predloženi postupak).



Slika 3.

Sama pravilnost rasporeda temena uslovljena je potrebom da se programski generiše relativno veliki broj ulaznih podataka. Ispostavilo se da naš heuristički postupak daje u proseku *pet puta* brže odgovor na pitanje o pripadnosti tačke zadatom poligonu nego li postupak koji koristi funkciju *ATAN2* − to se i moglo očekivati ako se zna da se za izračunavanje trigonometrijskih funkcija (kakva je i sistemska funkcija *ATAN2*) koriste *redovi*, što svakako povećava broj numeričkih operacija.

Završne napomene. Ipak, sintetička priroda (u odnosu na anvelopu aviona ili motora određenu sa svega nekoliko desetina tačaka) ovakvog primera uticala je da ovde predloženi algoritam ostane po strani sve dok nedavno saznanje − u nadasve sugestivnom razgovoru sa prof. dr Aleksandrom Lipkovskim (*Matematički fakultet*, Beograd) koji je ukazao na aktuelnost razmatranog problema pripadnosti tačke poligonalnoj oblasti za teoriju prepoznavanja oblika − nije stvorilo uverenje da bi takav postupak mogao da bude od koristi za širi krug istraživača i za neke

Buduće aktivnosti. Naime, imajući, s jedne strane, u vidu da se najrazličitije figure u ravni po pravilu opisuju poligonima obrazovanim od *velikog* broja *samo horizontalnih i vertikalnih* segmenata, a znajući, sa druge strane, da se u opisanom postupku *izračunava samo* ordinata y_j^{cross} preseka prave $x = x_P$ i nekih stranica poligona, jasno je da se za takve poligonalne oblasti ni to *jedino* izračunavanje ne mora sprovoditi (jer je za horizontalnu duž ta ordinata preseka (ako on postoji) *poznata*, a za vertikalnu se *izjednačava* sa ordinatom jednog od njenih temena), što bi svakako povećalo brzinu rada algoritma. Korišćenje naprednijih programa za pretraživanja koja se u ovom postupku vrše samo bi doprinelo tom ubrzavanju rada.

Literatura

[1] GRUPA AUTORA: Programski paket *MOBA* (*MOtorska BAza*), Žarkovo, int. izveštaj VTI ***, 1990.

[2] JORGOVIĆ,M. *O poligonima.* Matematička biblioteka 22, Beograd, Zavod za izdavanje udžbenika, 1962.

[3] ANSI: Computer Graphics - Graphical Kernel System (*GKS*) Functional Description, American National Standards Institute, 1985.

[4] GRIES,D. *The Science of Programming.* New York, Springer-Verlag, 1987.

[5] DREYFUS,M., GANGLOFF,C.: *La pratique du FORTRAN.* Paris, Dunod, 1975. (*prevod na ruski objavljen u Moskvi 1978.*)

[6] LIPKOVSKI,A. *Paket programa MORINO.* Žarkovo, int. izveštaj VTI ***, 1987.

Rad primljen: 20.4.2000.god.

* Kao što se vidi, komentarisani su i mogući naročiti slučajevi kada je bar jedna od stanica između kojih leži zadata tačka klase 0 (tj. bar jedna je upravna na *x*-osu) i pri tom se radi o *susednim* stranicama poligona (a što nije posebno navođeno u šemi 1).