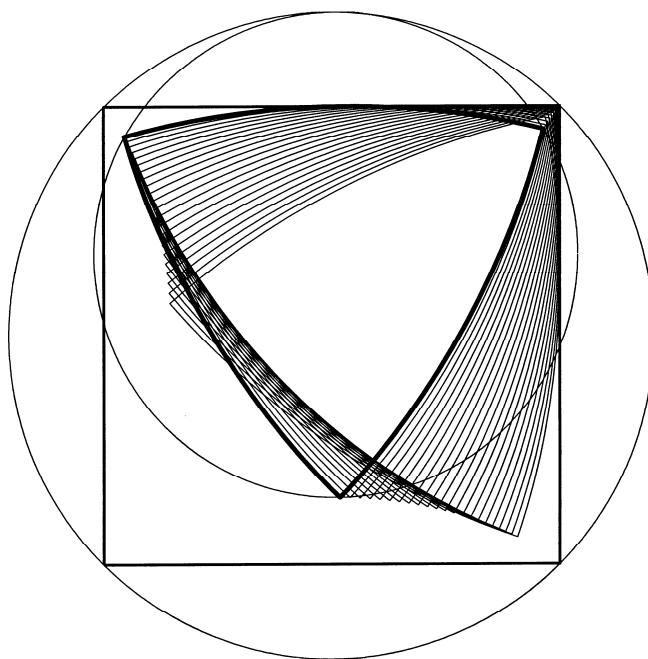


Mihăiță HORODINĂ

SIMULAREA GENERĂRII CURBELOR
PLANE PRIN RULARE.
APLICAȚII.



Iași 2013

Prefață

Lucrarea își propune să valorifice potențialul oferit de calculatorul personal în direcția demersului legat de asimilarea teoretică a procesului de generare prin rulare a curbelor plane (2D) pe baza simulării asistate de calculator folosind mediul Matlab. Lucrarea este concepută ca suport didactic auxiliar pentru cursul și aplicațiile disciplinei *Proiectarea Mașinilor-unelte*, parcursă de studenții din ciclul de licență specializarea *Mașini-unelte și Sisteme de Producție*, (Facultatea de Construcții de Mașini și Management Industrial din Universitatea Tehnică din Iași), dar poate fi punctual utilă și pentru o serie de alte discipline și specializări înrudite. Sunt tratate teoretic și exemplificate concret prin simulare (până la emularea programelor și ilustrarea grafică a rezultatelor execuției acestora) cele mai cunoscute procedee de definire prin rulare a curbelor plane, utilizate drept curbe generatoare în definirea suprafețelor complexe, pe organe de mașini uzuale (de tip arbore sau alezaj: roți dințate, roți de curea, roți de transmisii cu lanț, roți de clichet, arbori canelați, piese cu secțiune transversală poligonală, etc.). Procesul de rulare este privit ca având caracter dual: implicat în definirea tăișului de sculă și în transferul acestuia pe piesa finită. Fiecare etapă a demersului este finalizată printr-un set de programe de calculator cu funcționalitate demonstrată, accesibile studentului (sau cititorului interesat) pentru aprofundare, exercițiu și fixarea cunoștințelor. Toate programele utilizate sunt redate în anexele lucrării, și pot fi rulate ca atare. Sunt valorizate elementele esențiale de interdisciplinaritate din programul de pregătire al studenților și din planul de învățământ. Se regăsesc aici prin maniera de abordare/tratare sau prin finalitate o serie de noțiuni, concepte, proceduri și tehnici introduse la multe dintre disciplinele deja parcurse de către studenți. Autorul așteaptă eventuale elemente de *feed-back* și stă la dispoziția cititorilor interesați pe adresele: horodincea@tuiasi.ro sau hmihaita@yahoo.com. Slavă lui Dumnezeu pentru toate!

Mihăiță HORODINĂ

Iași, noiembrie 2013

Cuprins

1. Introducere	8
2. Aspecte teoretice ale generării curbelor plane prin rulare cerc-dreaptă (rulanta un cerc, baza o dreaptă).....	8
2.1 Elemente de principiu ale simulării asistate de calculator a generării prin rulare a curbelor plane (rulanta un cerc, baza o dreaptă).....	12
2.2 Finalitate practică a simulării asistate de calculator a generării prin rulare cerc pe dreaptă a curbelor plane... 15	15
2.3 Aspecte teoretice ale generării curbelor plane prin rulare dreaptă pe cerc.....	16
2.3.1 Elemente de principiu ale simulării asistate de calculator a generării prin rulare a danturii evolventice (rulanta o dreaptă, baza un cerc)	18
2.3.1.1 Elemente de utilitate practică ale simulării asistate de calculator a generării prin rulare a danturii evolventice.....	21
3. Aspecte teoretice ale simulării generării prin rulare a contururilor poligonale închise (rulanta dreaptă, baza cerc).	34
3.1 Limite ale generării prin rulare (rulanta dreaptă, baza cerc) a curbelor plane oarecare închise.....	39

4. Elemente de <i>feed-back</i> . Determinarea profilului cuțitului pieptene de mortezat roți dințate cu profil evolventic.....	51
5. Alte aplicații ale generării prin rulare dreaptă pe cerc.....	56
5.1 Generarea prin rulare dreaptă pe cerc a roților transmisiilor cu lanț.....	56
5.2 Generarea prin rulare dreaptă pe cerc a roților mecanismelor cu clichet	74
5.2.1 Simularea profilului transversal al roții de clichet.....	75
5.2.2 Obținerea profilului generator al roții de clichet.....	78
5.2.3 Validarea simulării. Generarea profilului transversal al roții de clichet prin rulare dreaptă pe cerc.....	83
6. Aspecte ale simulării rulării cerc pe cerc.....	88
6.1 Definierea profilului generator.....	89
6.2 Obținerea profilului de generat prin rularea profilului generator.....	98
7. Aspecte ale simulării rulării cerc în cerc (profil generator pe sculă de tip arbore).....	108
8. Aspecte ale simulării rulării cerc în cerc (profil generator pe sculă de tip alezaj).....	129
9. Concluzii.....	146
Bibliografie	148
Anexa 1.....	150
Anexa 2.....	152

Anexa 3.....	153
Anexa 4.....	156
Anexa 5.....	158
Anexa 6.....	160
Anexa 7.....	162
Anexa 8.....	165
Anexa 9.....	168
Anexa 10.....	170
Anexa 11.....	172
Anexa 12.....	175
Anexa 13.....	178
Anexa 14.....	181
Anexa 15.....	184
Anexa 16.....	188
Anexa 17.....	189
Anexa 18.....	191
Anexa 19.....	194
Anexa 20.....	197
Anexa 21.....	200
Anexa 21a.....	203
Anexa 22.....	206
Anexa 22a.....	210
Anexa 23.....	214
Anexa 24.....	217
Anexa 24a.....	220
Anexa 25.....	223

Anexa 26.....	227
Anexa 27.....	231
Anexa 28.....	236
Anexa 29.....	240
Anexa 30.....	244
Anexa 31.....	247
Anexa 32.....	251
Anexa 33.....	255
Anexa 34.....	259
Anexa 35.....	263
Anexa 36.....	265

1. Introducere

Rularea este un procedeu de generare cinematică a curbelor plane sau spațiale, utilizabile ca generatoare sau directoare în realizarea prin așchiere a suprafețelor pieselor pe mașini-unelte. Este în general utilizată în realizarea roților dințate cu flancuri evolventice sau neevolventice, a arborilor canelați cu flancuri drepte, paralele, neparalele sau evolventice, a roților de clichet și în general a tuturor pieselor cu simetrie rotațională.

Scopul acestui demers este de a explora analitic și a verifica sistematic prin simulare grafică asistată de calculator aspectele definitorii ale generării prin rulare a curbelor plane simple și compuse închise.

2. Aspecte teoretice ale generării curbelor plane prin rulare cerc-dreaptă (rulanta un cerc, baza o dreaptă)

Rularea plană [12] presupune existența a două curbe (de regulă închise) tangente în contact punctual (cu normală comună în punctul de contact). De regulă una dintre curbe (numită și *bază*, conform figurii 1) este fixă, cealaltă (numită și *rulantă*) execută o mișcare complexă cu cel puțin două componente, încât definește centru instantaneu de rotație în punctul de contact. (\mathbf{P} sau \mathbf{P}' pe figura 1). Dacă viteza relativă în punctul de contact este nulă se realizează rulare fără alunecare. Înfășurătoarea (raportată la bază) a pozițiilor succesive ale unei figuri

sau curbe solidare cu rulantă (**Csr**, pe figura 1) constituie curba generată prin rulare **Cgr** (solidară cu baza, pe figura 1).

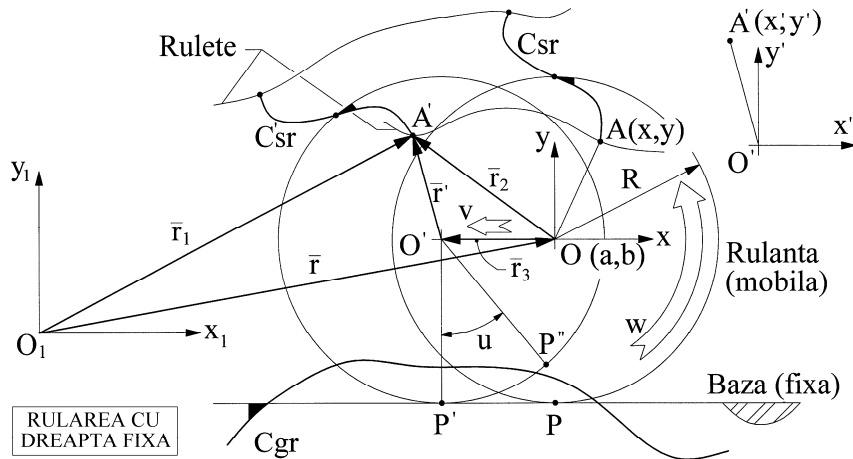


Figura 1. Elemente grafice conceptuale privind rularea cu rulantă circulară și bază rectilinie.

Cea mai facilă ilustrare a principiilor rulării se poate face conform figurii 1, cu rulantă cerc de rază R și bază rectilinie. Rulanta execută rotație cu viteza unghiulară w (în jurul centrului O) și translație cu viteza v (la nivelul centrului). Punctul P sau P' este centru instantaneu de rotație dacă este îndeplinită condiția cinematică:

$$(1) \quad v = w \cdot R$$

Relația (1) poate fi transpusă geometric după cum urmează. Când rulantă își deplasează centrul din O în O' , centrul instantaneu de rotație trece din P în P' (cu P'' amprenta lui P pe rulantă). În aceste condiții lungimea arcului $P'P''$ este egală cu cea a segmentului PP' respectiv a segmentului OO' , adică:

$$(2) \quad OO' = R \cdot u$$

unde u este unghiul corespunzător arcului $P'P''$ exprimat în radiani.

Fie Csr și Csr' cele două poziții ale curbei solidare cu rulanta, și $A(x,y)$ respectiv $A'(x',y')$ un punct generic al acesteia, cu coordonatele exprimate în raport cu sistemul de coordonate xOy respectiv $x'O'y'$. Aici $x'O'y'$ este noua poziție a reperului xOy când rulanta s-a rotit cu unghiul u , obținută prin translația în lungul abscisei care păstrează paralelismul cu abscisa sistemului de coordonate $x_1O_1y_1$, exterior, fix, solidar cu baza. Determinarea expresiei analitice (parametrice) a coordonatelor (x_1, y_1) ale punctului generic A' în $x_1O_1y_1$ permite determinarea prin calcul asistat și reprezentarea grafică a pozițiilor tuturor punctelor curbelor Csr' pentru orice poziție a rulantei.

Din considerente geometrice expuse pe figura 1, poziția punctului A' în sistemul de coordonate $x_1O_1y_1$ poate fi descrisă cu ajutorul ecuației vectoriale:

$$(3) \quad \vec{r}_1 = \vec{r} + \vec{r}_2$$

sau:

$$(4) \quad \vec{r}_1 = \vec{r} + \vec{r}_3 + \vec{r}'$$

Proiecția acestei ecuații vectoriale pe axele sistemului $x_1O_1y_1$ permite determinarea ecuațiilor parametrice ale traiectoriei punctului A (A'), numită în practică și ruletă (sau curbă cicloidală).

Este interesant de remarcat faptul că prin rulare se generează o familie de curbe Csr (Csr') precum și o familie de rulete. Înfașurătoarea raportată la bază a ambelor familii este aceeași curbă generată prin rulare Cgr . Acest aspect va rezulta și din prezentările grafice ale simulării rulării.

În ecuația (4) singurele proiecții neevidente sunt cele ale vectorului \vec{r}' în sistemul $x_1O_1y_1$, aceleași cu proiecțiile în sistemul $x'O'y'$ (din cauza paralelismului axelor). Poziția (x', y') a punctului A' din $x'O'y'$ rezultă prin rotirea segmentului OA din xOy cu unghiul u (în jurul lui O) în sens trigonometric. Același rezultat se obține dacă se păstrează fixe punctele O și A dar se rotește reperul xOy în jurul lui O (în $x'O'y'$), cu unghiul u în sens invers trigonometric (orar), conform

figurii 2. Problema se reduce la o simplă schimbare a sistemului de

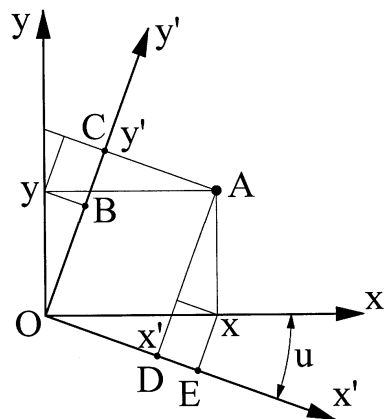


Figura 2. Deducerea expresiei analitice a coordonatelor punctului A în sistemul $x'O'y'$.

coordonate. Din exprimarea segmentului $OD=OE-DE$ rezultă abscisa x' , din exprimarea segmentului $OC=OB+BC$ rezultă ordonata y' conform cu:

$$(5) \quad \begin{aligned} x' &= x \cdot \cos(u) - y \cdot \sin(u) \text{ și} \\ y' &= y \cdot \cos(u) + x \cdot \sin(u) \end{aligned}$$

În (5) u este parametrul ecuațiilor. Cu titlu de generalizare, dacă se consideră rotația reperului xOy în sens trigonometric, în (5) se consideră semnul minus pentru parametrul u .

Cu aceasta se pot scrie proiecțiile ecuației vectoriale (4) conform cu:

$$(6) \quad x_I = a - R \cdot u + x' \quad \text{și} \quad y_I = b + y'$$

Sau, în extenso, ținând seama de (5) :

$$(7) \quad x_I = a - R \cdot u + x \cdot \cos(u) - y \cdot \sin(u)$$

$$(8) \quad y_I = b + y \cdot \cos(u) + x \cdot \sin(u)$$

S-au obținut astfel ecuațiile parametrice ale punctului curent al curbei Csr corespunzătoare rulării cu rulantă cerc de rază R și baza o dreaptă, aplicabile oricărui punct solidar cu rulantă.

2.1 Elemente de principiu ale simulării asistate de calculator a generării prin rulare a curbelor plane (rulanta un cerc, baza o dreaptă)

Finalitatea demersului anterior poate fi asigurată, ilustrată și completată cu utilizarea calculatorului pentru simularea rulării, pe baza reprezentărilor grafice corespunzătoare. S-au utilizat facilitățile oferite de mediul de programare Matlab [1, 7].

Se ilustrează în continuare simularea rulării cu utilizarea unei rulante cerc de rază 10 mm. Se propune drept figură plană *Csr* atașată rulantei un triunghi echilateral înscris în cercul rulantei. Secvența de program descrisă pe baza *listing*-ului de mai jos permite ilustrarea pozițiilor succesive ale triunghiului precum și a înfășurătorii raportate la bază a acestora, corespunzător unei rotații complete a rulantei.

PROGRAM TRIUNGHI

```
L1/r=10;a=100;b=0;x(1)=0;y(1)=10;x(2)=8.660254;y(2)=-
5;x(3)=-8.660254;
y(3)=-5;x(4)=0;y(4)=10;k=4;
L2/l=1;
L3/for i=0:2*pi/50:2*pi;u=i;
L4/for j=1:k;
    L5/x1(l)=a-r*u+x(j)*cos(u)-y(j)*sin(u);
    L6/y1(l)=b+x(j)*sin(u)+y(j)*cos(u);l=l+1;
L7/end
L8/end
L9/l=l-1;
L10/plot(x1,y1,'k');hold on;axis equal;maxx=-
100;minx=100;
L11/for i=1:l;
    L12/if x1(i)>maxx;maxx=x1(i);else end
    L13/if x1(i)<minx;minx=x1(i);else end
    L14/x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
L15/clear x1;clear y1;x1=x2;y1=y2;k=1;
```

```

L16/for i=1:l-1;
    L17/dif=(x1(i+1)-x1(i));if
dif==0;dif=0.0000000001;else end
    L18/m(k)=(y1(i+1)-y1(i))/dif;n(k)=y1(i)-m(k)*x1(i);
    L19/k=k+1;
L20/end;k=k-1;
L21/inc2=(maxx-minx)/3000;l=1;minminy=10000;
L22/for i=minx+0.001:inc2:maxx-inc2;miny=1000;ics=0;
    L23/abs(l)=i;
    L24/for j=1:k-1;
        L25/if i>x1(j);if i<x1(j+1);int=m(j)*i+n(j);int;if
L26/int<miny;ord(l)=int;miny=int;else end;else end;else
end
        L27/if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);int;
L28/ if int<miny;ord(l)=int;miny=int;else end;else
end;else end
L29/end;l=l+1;
L30/if miny<minminy;minminy=miny;else end
L31/end
L32/dim1=size(ord);dim=dim1(2)
L33/plot(abs,ord,'k','LineWidth',1.5);hold on
L34/axis equal
L35/ord=minminy-ord;
L36/plot(abs,ord,'k','LineWidth',1.5)

```

Pentru comentarii, marcarea numărului liniei curente i s-a făcut cu notația $L_i/$ (eliminată în cazul rulării efective în Matlab). Pentru simplificarea programului (reluat în Anexa 1) și reducerea timpului de execuție s-au considerat numai coordonatele (x , y) ale vârfulor triunghiului (poziția segmentelor ce descriu laturile triunghiului este definită numai de acestea), conform liniei $L1/$. Matricile \mathbf{x} și \mathbf{y} conțin câte patru elemente (în prima și ultima locație fiind înscrise aceleași valori) pentru a se realiza descrierea unei contur triunghiular închis.

Ciclul de instrucțiuni de la $L2/$ la $L8/$ realizează calculul coordonatelor de descriere a pozițiilor succesive ale vârfulor triunghiului pentru fiecare variație a unghiului u egală cu $2\pi/50$ radiani. Numitorul expresiei poate fi evident modificat, obținându-se un număr mai mare de poziții succesive ale triunghiului. Liniile $L5/$ și $L6/$ realizează calculele coordonatelor conform ecuațiilor (7) și (8). În

matricile $\mathbf{x1}$ și $\mathbf{y1}$ se găsesc coordonatele segmentelor de descriere a segmentelor corespunzătoare fiecărei laturi a triunghiului, pentru toate pozițiile acestuia. Aici $\mathbf{1}$ este variabila contor, cu titlu de exemplu având valoarea 204 la ieșirea din ciclu. Reprezentarea grafică a pozițiilor succesive ale triunghiului corespunzător rulării se face cu

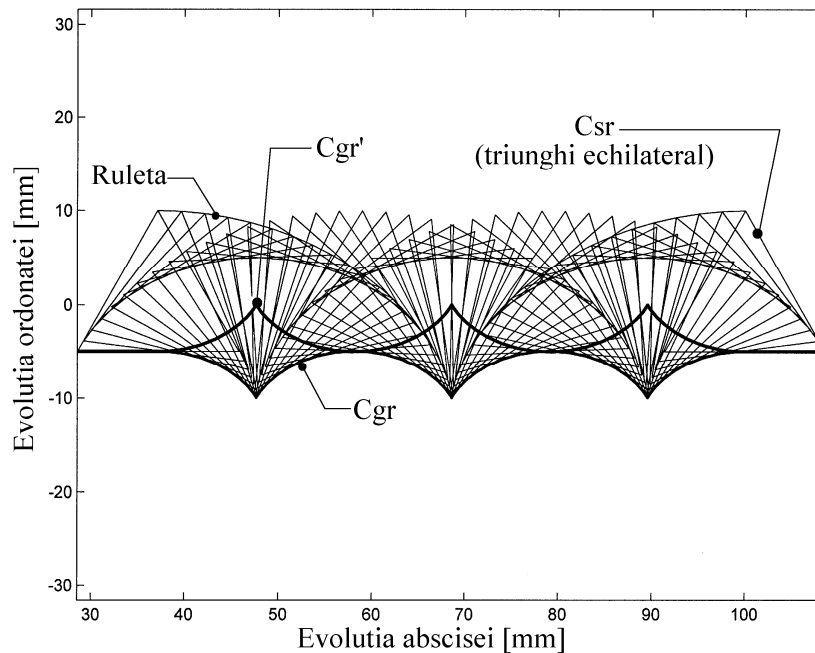


Figura 3. Elemente grafice ale simulării rulării cerc-dreaptă cu descrierea înfășurătorii pozițiilor succesive ale unui triunghi echilateral.

ajutorul primei instrucțiuni din linia L5/, conform figurii 3.

Cu ajutorul grupului de instrucțiuni dintre L11/ și L15/ realizează determinarea valorilor extreme (**maxx** și **minx**) ale absciselor din matricea $\mathbf{x1}$ și se face rotația matricilor $\mathbf{x1}$ și $\mathbf{y1}$ (valorile din $\mathbf{x1}$ apar astfel în ordine crescătoare).

Cu ajutorul setului de instrucțiuni dintre L16/ și L20/ se realizează calculul parametrilor de definiție a dreptelor suport (panta $\mathbf{m(k)}$ și tăietura $\mathbf{n(k)}$) pentru segmentele ce definesc pozițiile succesive ale laturilor triunghiului (se folosesc coordonatele a câte două vârfuri succesive).

Cu ajutorul setului de instrucțiuni dintre L21/ și L31/ se realizează determinarea descrierii sintetice a înfășurătoarei raportate la bază a pozițiilor succesive ale triunghiului în mișcarea de rulare (curba **Cgr**). Se generează un set virtual de drepte verticale echidistante (cu distanța dintre ele $inc2 = (maxx - minx) / 3000$). Pentru fiecare dreaptă se calculează coordonatele punctelor de intersecție cu drepte suport ale laturilor triunghiului care îndeplinesc următoarea condiție: punctul de intersecție aparține unui segment de descriere a respectivei drepte. Coordonatele punctului de intersecție rezultă din rezolvarea sistemului de ecuații:

$$(9) \quad y = m \cdot x + n$$

$$(10) \quad x = i$$

Dintre toate punctele de intersecție se rețin coordonatele celui care are ordonată minimă. Coordonatele acestui punct se copie în matricile **abs** și **ord**. Cu instrucțiunea din linia L33/ se realizează reprezentarea grafică a înfășurătoarei **Cgr** (conform figurii 3).

Cu ajutorul instrucțiunilor L35/ și L36/ se generează o curbă **Cgr'** imagine în oglindă a **Cgr** conform figurii 3. Punctele de maxim ale acesteia au ordonata nulă. Rațiunea generării acesteia va rezulta ulterior.

2.2 Finalitate practică a simulării asistate de calculator a generării prin rulare cerc pe dreaptă a curbelor plane

Cea mai interesantă finalitate a acestui demers este legată de sinteza profilului de tăiș de sculă de mortezat de tip pieptene [13] pentru generarea prin rulare a curbelor generatoare la prelucrarea prin așchiere [5, 6, 8]. Prin rularea cu rulantă dreaptă și baza cerc (rularea dreaptă pe cerc) curba **Cgr** sau **Cgr'** (generată anterior, solidară cu rulantă) generează pe bază curba **Csr**. Aceasta se definește fie ca înfășurătoare a traiectoriilor succesive ale curbei **Cgr** (sau **Cgr'**) fie a

ruletelor generate de aceasta (ca familie de curbe cicloidale). Această finalitate practică va fi prezentată ulterior în orizontul figurii 13.

2. 3. Aspecte teoretice ale generării curbelor plane prin rulare dreaptă pe cerc

În practica generării prin rulare a danturii roților dințate s-a consacrat utilizarea sculelor derivate din cremaliera generatoare (de exemplu cuțitul cremalieră de mortezat sau freza melc-modul). Găsim

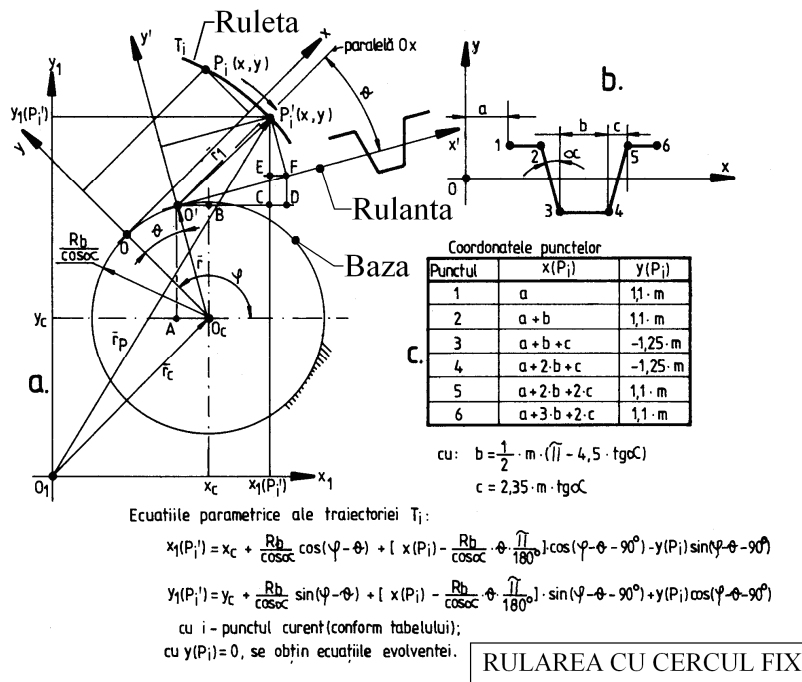


Figura 4. Aspecte teoretice ale generării prin rulare a curbelor plane (rulanta dreaptă și baza cerc). Aplicație în cazul generării danturii roților dințate cu profil evolventic.

astfel cea mai bună oportunitate de simulare în contextul lucrării, cu elementele de principiu expuse în figura 4 [9, 10]. Se definește mai întâi un dinte al cremalierii generatoare în sistemul de coordonate xOy

(conform figurii 4.b) prin coordonatele punctelor de definire a segmentelor tăişurilor, cu valorile precizate în tabelul din figura 4.c. Aceste valori rezultă din uzanțele de definire a cremalierii generatoare, cu m - modulul danturii, α - unghiul de angrenare (uzual 20°), $\pi \cdot m$ - pasul cremalierii (egal cu pasul circumferențial al danturii pe cercul de rostogolire de rază $R_b / \cos \alpha = m \cdot Z / 2 = R_r$, aceeași cu raza bazei).

În formula de mai sus R_b se numește rază de bază, Z este numărul de dinți ai roții dințate de generat.

Conform figurii 4, fie sistemul fix $x_1O_1y_1$ solidar cu baza. Fie un punct oarecare P_i atașat sistemului xOy . Fie sistemul xOy atașat bazei în O , cu tangența abscisei la cercul de rostogolire. Să considerăm comportarea ca rulantă a acestei abscise. Fie trecerea sistemului xOy în poziția $x'O'y'$, deci a punctului P_i în poziția P_i' , prin rularea în condițiile de mai sus, cu rotirea axei ordonatei cu unghiul θ , păstrarea ordonatei și reducerea abscisei cu mărimea arcului de cerc $OO' = R_r \cdot \theta$, cu exprimarea lui θ în radiani. Aceasta înseamnă că orice figură plană din sistemul xOy (deci și a profilului de dinte al cremalierii din figura 4 b.) realizează simultan rotația odată cu sistemul și translația în lungul abscisei. Traectoria T_i a punctului P_i este evident o ruletă, ecuațiile parametrice ale acesteia rezultă din ecuația vectorială (11) scrisă referitor la poziția punctului P_i :

$$(11) \quad \overline{r_P} = \overline{r_C} + \overline{r} + \overline{r_I}$$

Proiecția acestei ecuații pe abscisa respectiv ordonata sistemului $x_1O_1y_1$ conduce la obținerea ecuațiilor parametrice ale ruletei T_i conform cu:

(12)

$$x_i(P_i') = x_C + R_r \cdot \cos(\varphi - \theta) + [x(P_i) - R_r \cdot \theta] \cdot \cos(\gamma) - [y(P_i)] \cdot \sin(\gamma)$$

(13)

$$y_i(P_i') = y_C + R_r \cdot \sin(\varphi - \theta) + [x(P_i) - R_r \cdot \theta] \cdot \sin(\gamma) + [y(P_i)] \cdot \cos(\gamma)$$

În (12) și (13) s-a folosit notația $\gamma = \varphi - \theta - \pi/2$, cu γ , φ și θ unghiuri exprimate în radiani. Parametrul ecuațiilor este unghiul θ . În spațiul

figurii 4 sunt prezentate aceleași două ecuații dar cu exprimarea unghiurilor în grade hexazecimale.

Aplicând setul de ecuații fiecăruia dintre punctele din figura 4 b., cu coordonatele prezentate în figura 4 c. se poate descrie prin calcul și reprezenta grafic în sistemul $x_1O_1y_1$ pozițiile succesive ale dintelui cremalierii. Infășurătoarea acestor poziții succesive descrie într-o zonă limitată a sa un gol de dinte generat pe roata dințată. Evident că este de așteptat ca extinzând cremaliera la un număr suficient de mare de dinți (prin extrapolarea coordonatelor din figura 4 c.) și o valoare maximă a unghiului $\theta = \pi$ să se genereze grafic o roată dințată completă. Acest aspect va fi ilustrat în paragraful următor.

2. 3.1 Elemente de principiu ale simulării asistate de calculator a generării prin rulare a danturii evolventice (rulanta dreaptă, baza cerc)

Pe baza considerațiilor de mai sus s-au întreprins demersuri de simulare grafică asistată de calculator a generării prin rulare a danturii evolventice. Pe baza programului prezentat mai jos (reluat în Anexa 2) a fost obținută simularea generării danturii din figura 5.

PROGRAM SIMGENDANT

```
L1/clear all;close all;m=25;z=20;a=120;fi=90*pi/180;
alfa=20*pi/180;c=2.35*m*tan(alfa);b=0.5*m*(pi-
4.7*tan(alfa));dep=0*m;
rr=0.5*m*(z);x(1)=a;x(2)=a+b;x(3)=a+b+c;x(4)=x(3)+b;x(5)=
x(4)+c;x(6)=x(5)+b;
```

```
L2/y(1)=1.1*m+dep;y(2)=1.1*m+dep;
y(3)=-1.25*m+dep;y(4)=-1.25*m+dep;
y(5)=1.1*m+dep;y(6)=1.1*m+dep;
```

```
L3/ ndinti=z+5;for j=1:ndinti;for
i=1:6;x(j*6+i)=x(i)+j*m*pi;
y(j*6+i)=y(i);end;end; xc=0;yc=0;
```

```

L4/liminf teta=0;limsupteta=2.5*pi;inc=2*pi/100;
L5/for teta=liminf teta:inc:limsupteta;
  L6/for i=1:6*(ndinti)
    L7/x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
    L8/y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-

```

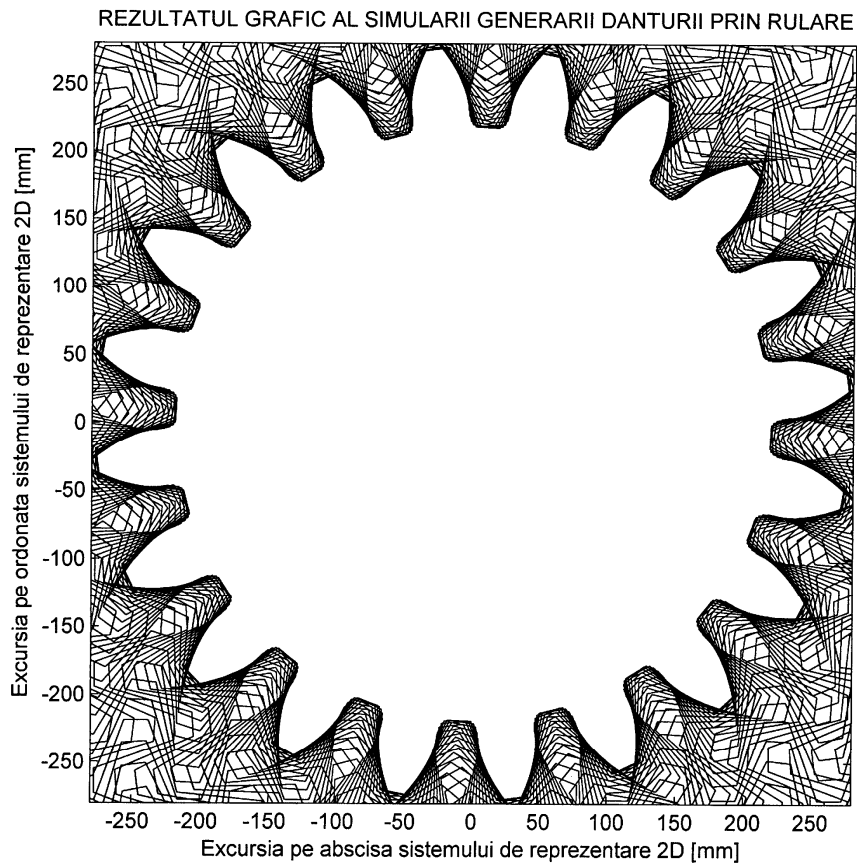


Figura 5. Rezultat grafic al simulării generării danturii evolventice prin rulare (bază cerc și rulantă dreaptă) cu utilizarea programului *SIMGENDANT*.

```

teta-pi/2)+y(i)*cos(fi-teta-pi/2);
  L9/end
  L10/plot(x1,y1,'k');hold on;axis equal
L11/end

```

```
L12/axis([- (rr+1.2*m), (rr+1.2*m), - (rr+1.2*m), (rr+1.2*m) ])
Title('REZULTATUL GRAFIC AL SIMULARII GENERARII DANTURII
PRIN RULARE')
xlabel('Excursia pe abscisa sistemului de reprezentare 2D
[mm]')
ylabel('Excursia pe ordonata sistemului de reprezentare
2D [mm]')
```

Pe figura 5 s-a obținut reprezentarea grafică clară a unei roți dințate definite prin înfășurătoarea pozițiilor succesive ale cremalierii generatoare. Se obține rapid o confirmare rapidă asupra corectitudinii demersului teoretic anterior, validare indispensabilă continuării modelării.

În linia L1/ se face prescrierea parametrilor necesari definirii rulantei (modul, număr de dinți, rază de rulare), a parametrilor necesari definirii geometrice a dintelui cremalierii precum și abscisele punctelor de definire a segmentelor ce delimitează tăișurile (conform figurii 4.c). În linia L2/ se face prescrierea valorilor ordonatelor acestor puncte. S-a introdus suplimentar variabila **dep** prin care poate fi controlată deplasarea de profil (pozitivă sau negativă).

În linia L3/ se extrapolează prin calcul numeric coordonatele de definire ale unui dinte la întreaga cremalieră (considerând că aceasta are Z+5 dinți).

În linia L4/ se fixează limitele de variație ale parametrului θ (acoperitor unei rotații cu $2\cdot\pi$ plus un spațiu unghiular care să asigure definirea completă a danturii) și incrementul de variație al acestuia ($2\cdot\pi/100$). Numitorul acestui increment este evident modificabil.

Între liniile L5/și L11/ se definesc două cicluri îmbricate (unul în interiorul celuilalt) prin care se asigură pe baza ecuațiilor (12) și (13) calculul coordonatelor punctelor de definiție a tăișurilor întregii cremalierii (pe baza liniilor L7/și L8/) pentru toate cele 100 poziții ale acesteia, corespunzător rulării.

Valorile coordonatelor acestor puncte sunt stocate în matricile **x1** respectiv **y1** și folosite la reprezentarea grafică a cremalierii pentru fiecare poziție (cu instrucțiunea **plot...** din L10/).

Cu ajutorul instrucțiunii din L12/ se realizează limitarea zonei de reprezentare grafică la perimetrul roții.

2. 3.1.1 Elemente de utilitate practică ale simulării asistate de calculator a generării prin rulare a danturii evolventice

Primul obiectiv al simulării a fost ilustrarea practică a fenomenului de generare a danturii evolventice prin rulare. Se pune astfel la dispoziția cititorului un instrument de autoinstruire, verificare, corecție și optimizare a procesului de generare a danturii prealabil transferului pe sistemul de materializare industrială (mașina de danturat). Evident că prelucrarea roții dințate presupune utilizarea unei scule cremalieră de mortezat cu mișcare principală, de așchiere, rectilie alternativă, perpendicular pe planul figurii sau a unei scule derivate (freze melc modul).

Demersul poate fi continuat prin completarea programului SIMGENDANT cu facilități de determinare automată a profilului teoretic complet al roții dințate (flancuri, fund și cap de dinte). Cu ajutorul programului DETPROF (reluat în Anexa 3) se realizează determinarea coordonatelor acestui profil.

PROGRAM DETPROF

```
clear all;close all;m=25;z=20;a=120;fi=90*pi/180;m1=m;
alfa=20*pi/180;c=2.35*m*tan(alfa);b=0.5*m*(pi-
4.7*tan(alfa));dep=0*m;
rr=0.5*m*(z);
x(1)=a;x(2)=a+b;x(3)=a+b+c;x(4)=x(3)+b;x(5)=x(4)+c;x(6)=x
(5)+b;
y(1)=1.1*m+dep;y(2)=1.1*m+dep;y(3)=-1.25*m+dep;y(4)=-
1.25*m+dep;y(5)=1.1*m+dep;y(6)=1.1*m+dep;ndinti=z+5;
for j=1:ndinti;for
i=1:6;x(j*6+i)=x(i)+j*m*pi;y(j*6+i)=y(i);end;end
xc=0;yc=0;
```

```

l=1;
liminf teta=0;limsup teta=2.5*pi;inc=2*pi/300+0.00001;
for teta=liminf teta:inc:limsup teta;
    for i=1:6*(ndinti)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);l=l+1;
    end

end;l=l-1;
%plot(x1,y1,'k');hold on;axis equal
axis([- (rr+1.4*m1), (rr+1.4*m1),- (rr+1.4*m1), (rr+1.4*m1)])
Title('REZULTATUL GRAFIC AL SIMULARII GENERARII DANTURII
PRIN RULARE')
xlabel('Excursia pe abscisa sistemului de reprezentare 2D
[mm]')
ylabel('Excursia pe ordonata sistemului de reprezentare
2D [mm]')

%rutina de determinare a profilului
k=1;
L1/for i=1:l-1;
    dif=(x1(i+1)-x1(i));if dif==0;dif=0.0000000001;else
end
    m(k)=(y1(i+1)-y1(i))/dif;n(k)=y1(i)-m(k)*x1(i);k=k+1;
L2/end; k=k-1;
L3/inc=2*pi/2000;l=1;
L4/for i=0:inc:2.1*pi;mindist=1000;
    L5/xa=xc;ya=yc;xb=xa+5*cos(i);yb=ya+5*sin(i);dif=xb-
xa;if L6/dif==0;dif=0.0000000001;else end;
    L7/mi=(yb-ya)/dif;ni=ya-xa*mi;
    L8/for j=1:k-1; dif1=mi-m(j);if
dif1==0;dif1=0.0000000001;else end;
        L9/intx=(n(j)-
ni)/dif1;inty=intx*m(j)+n(j);dist=(abs(intx-
xa)^2+abs(inty-ya)^2)^0.5;
        L10/dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        L11/if dist1<dist;
            L12/if intx==x1(j+1);if inty==y1(j+1);
L13/ if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;else
end;else end;else end;

```

```

L14/if intx==x1(j);if inty==y1(j);
L15/if
dist<mindist;absc(1)=intx;ord(1)=inty;mindist=dist;else
end;else
end;else end;
L16/if intx<x1(j+1);if intx>x1(j);if
dist<mindist;absc(1)=intx;ord(1)=inty;mindist=dist;else
end;else end;else end;
L17/if intx>x1(j+1);if intx<x1(j);if
dist<mindist;absc(1)=intx;ord(1)=inty;mindist=dist;else
end;else end;else end;

```

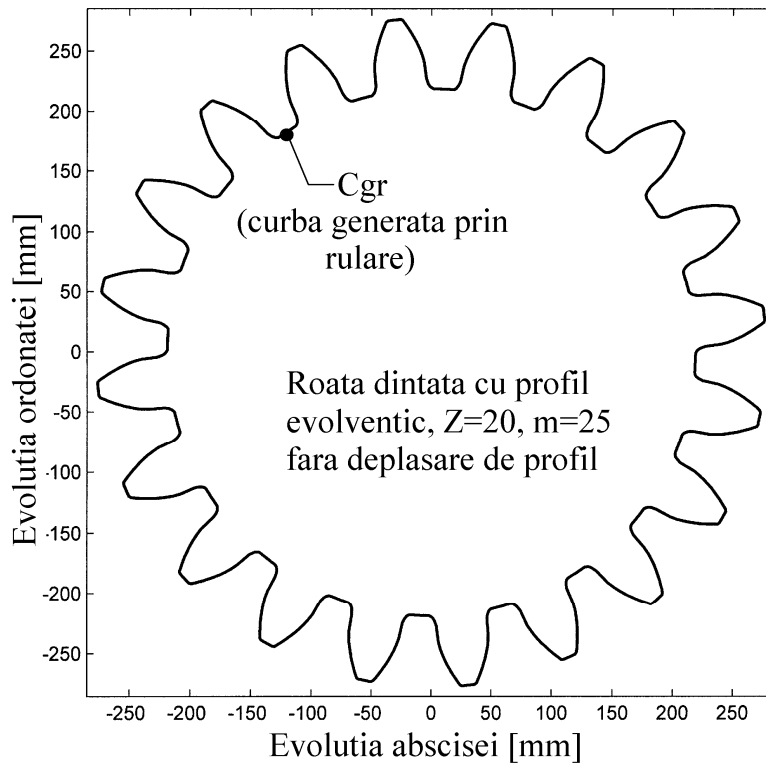


Figura 6. Descriere grafică a profilului roții dințate din figura 5 dedus prin calcul cu ajutorul programului DETPROF.

```

L18/if inty<y1(j+1);if inty>y1(j);if
dist<mindist;absc(1)=intx;ord(1)=inty;mindist=dist;else
end;else end;else end;

```

```

L19/if inty>y1(j+1);if inty<y1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;else
end;else end;else end;
L20/else end
L21/end;
L22/l=l+1;
L23/end
L24/plot(absc,ord,'k','LineWidth',1.5);axis equal;

```

Prima parte a programului (nemarkată cu bold) copie aproape identic programul SIMGENDANT. Excepție face maniera de memorare a coordonatelor punctelor de definiție a tășurilor cremalierii. În matricile **x1** respectiv **y1** sunt memorate coordonatele tuturor punctelor, pentru toate pozițiile cremalierii. De asemeni instrucțiunea de desenare **plot . . .** (acum suspendată) a fost scoasă în afara ciclurilor de calcul a coordonatelor.

Ultima parte a programului (marcată cu bold) realizează detecția sintetică a descrierii înfășurătoarei pozițiilor succesive ale cremalierii, adică a profilului roții dințate. În figura 6 se prezintă grafic profilul de roată dințată cu coordonatele (abscisa, ordonata) deduse prin calcul cu ajutorul programului DESPROF, ca înfășurătoare a pozițiilor succesive din figura 5.

În continuare se fac o serie de comentarii în legătură cu această ultimă parte. Ciclul dintre liniile L1/și L2/ realizează calculul și memorarea (în matricile **m** și **n**) valorilor pantelor și tăieturilor dreptelor suport ale tuturor tășurilor cremalierii pentru toate pozițiile acestora. Pentru datele de rulare utilizate s-au definit 49.499 drepte distincte.

Între L4/și L7/se definește o familie de drepte de test ce trec prin origine, au pantă variabilă și conțin un punct suplimentar. În liniile L8/și L9/se inițiază o procedură de calcul a coordonatelor de intersecție ale dreptei de test curente cu dreptele suport ale segmentelor de definiție ale tășurilor cremalierii. Se realizează o selecție succesivă a acestor puncte de intersecție. Sunt reținute în primul rând cele care aparțin segmentelor de definiție a tășurilor (inclusiv eventual cele ce coincid cu capetele acestora). Se selectează apoi numai cele ce aparțin semidreptei de test limitate de origine și conține punctul suplimentar

(cu instrucțiunea inițiată în L11/). Și, în sfârșit, formal cu instrucțiuni de tipul `if dist<mindist`, se selectează numai acele puncte de intersecție care sunt cele mai apropiate de origine. Aceste puncte aparțin profilului teoretic al roții dințate, cu coordonatele reținute în matricile `absc` și `ord`. Cu instrucțiunea din L24/ se realizează reprezentarea grafică din figura 6. Acuratețea descrierii profilului

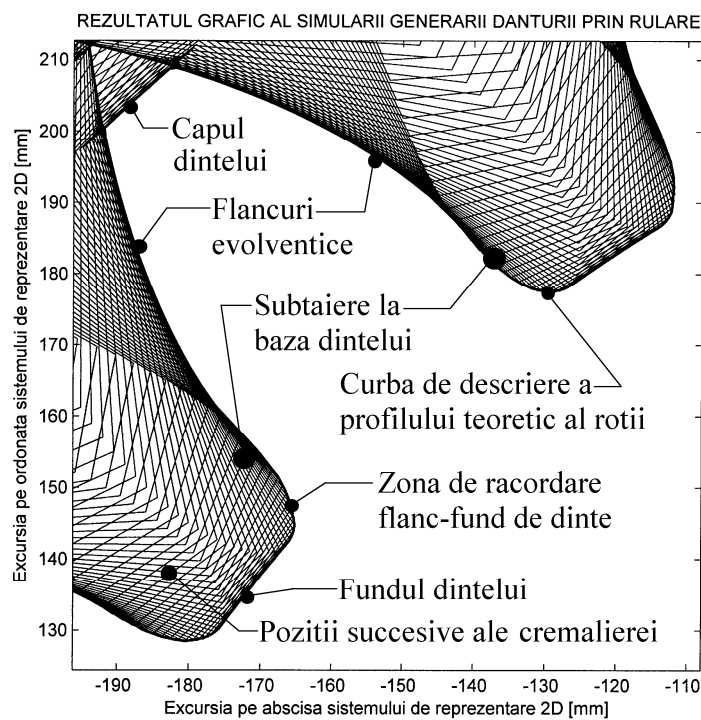


Figura 7. Detaliu asupra descrierii grafice a unui dinte al roții dințate generate cu ajutorul programului DETPROF.

depinde de valoarea incremenților unghiulari utilizați în simulare.

Profilul astfel rezultat poate fi folosit de exemplu pentru prelucrarea roții dințate pe o mașină-unealtă cu deplasare controlată 2D, cu ajutorul unei freze cilindro-frontală, sau, pentru roți dințate extrem de mici, prin utilizarea electroeroziunii cu fir sau cu fascicol Laser.

Un detaliu cu profilul teoretic al roții astfel dedus și cu pozițiile succesive ale cremalieriei este prezentat în figura 7. Pentru obținerea

acesteia s-a activat instrucțiunea `%plot(x1,y1,'k');hold on;axis equal` prin eliminarea semnului `%` și s-a apelat la selectarea unei ferestre grafice din reprezentarea astfel obținută. Incrementul unghiular al rotației cremalierii este de trei ori mai mic aici decât în figura 5. Intre elementele evidențiate pe figură semnalăm un ușor fenomen de subțaiere la baza dintelui [17] precum și zona de racordare flanc-fund de dinte. Această zonă de racordare apare datorită faptului că aici rularea se realizează cu viteză relativă nenulă. În practică zona de racordare nu este implicată în contactul roților, există un joc între capul și fundul roților conjugate.

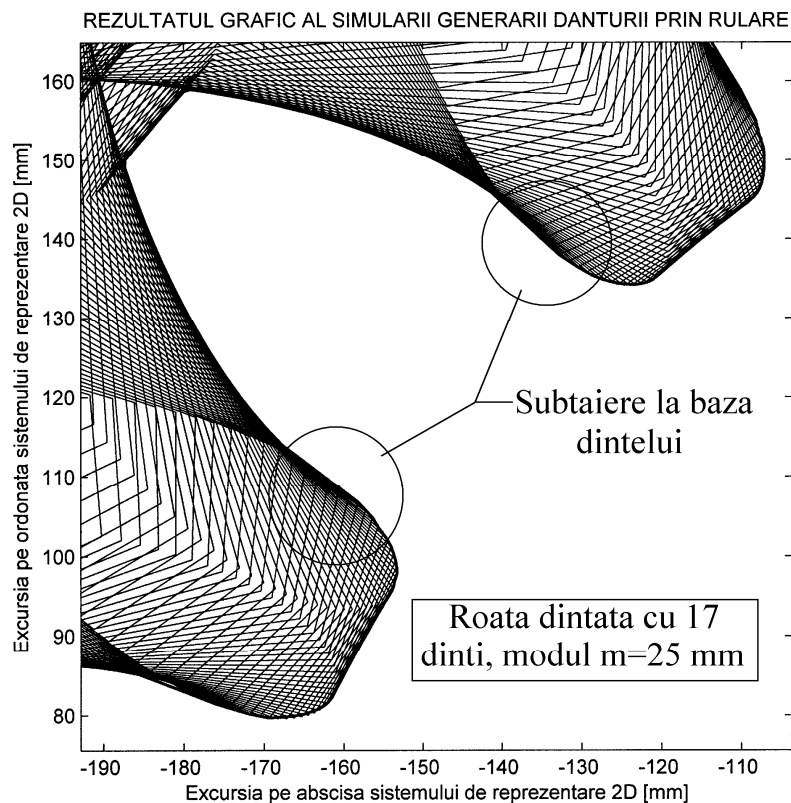


Figura 8. *Reliefarea fenomenului de subțaiere la baza dintelui pentru o roată dințată cu 17 dinți.*

În legătură cu fenomenul de subțaiere la baza dintelui în practica proiectării angrenajelor este binecunoscută soluția înlăturării

acesteia prin așa numita deplasare pozitivă de profil. Elementele de logică programată permit reliefarea prin simulare a acestei posibilități de corecție. Valoarea deplasării pozitive de profil este însă limitată de apariția unui alt fenomen, anume ascuțirea vârfului dintelui cu risc de rupere-strivire. Un compromis între aspectele nedorite ale subtăierii și

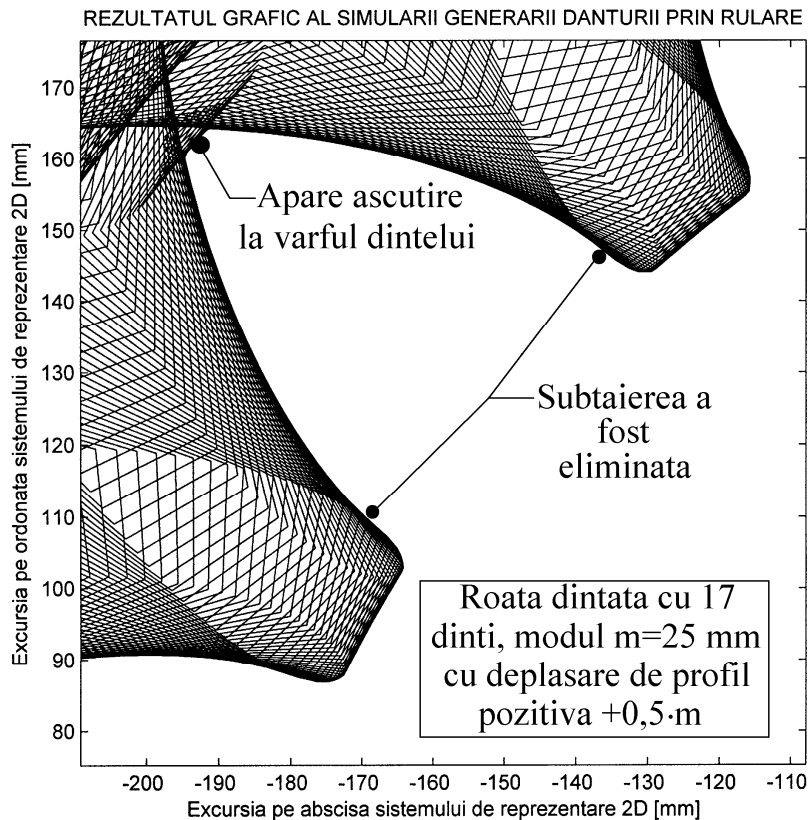


Figura 9. Reliefaarea eliminării subtăierii (prin deplasare de profil) și a apariției ascuțirii vârfului dintelui.

ale ascuțirii în condițiile deplasării de profil se poate asigura prin limitarea inferioară a numărului de dinți ai roții. Sub rezerva aplicării altor soluții de compromis, în practică se consideră limitarea la 17 a numărului de dinți. În figura 8 se prezintă un detaliu asupra unui dinte al unei roți cu 17 dinți generat în condiții identice cu cele din figura 7,

cu ajutorul programului DETPROF (în prima linie a acestuia s-a prescris valoarea $z=17$).

Comparativ cu figura 7 se observă o subțiere mai pronunțată la baza dintelui, fenomenul fiind cu atât mai accentuat cu cât scade numărul de dinți. Realizarea deplasării pozitive de profil se face prin translarea în sensul pozitiv al axei ordonatei a cremalierii generatoare în sistemul xOy (pe figura 4). Se propune simularea corecției subțierii prin utilizarea unei deplasări egale cu $0,5 \cdot m$. Pentru aceasta se rescrie instrucțiunea $dep=0 \cdot m$; din a doua linie a programului DETPROF ca $dep=0,5 \cdot m$; Rularea acestui program în aceleași condiții cu cele utilizate pentru generarea figurilor 7 și 8 conduce la rezultatul grafic din figura 9.

Este evidentă eliminarea subțierii dar și apariția fenomenului de ascuțire a vârfului dintelui, cu afectarea severă a capacității portante a dintelui roții.

Este binecunoscut faptul că în practică se utilizează roți cu

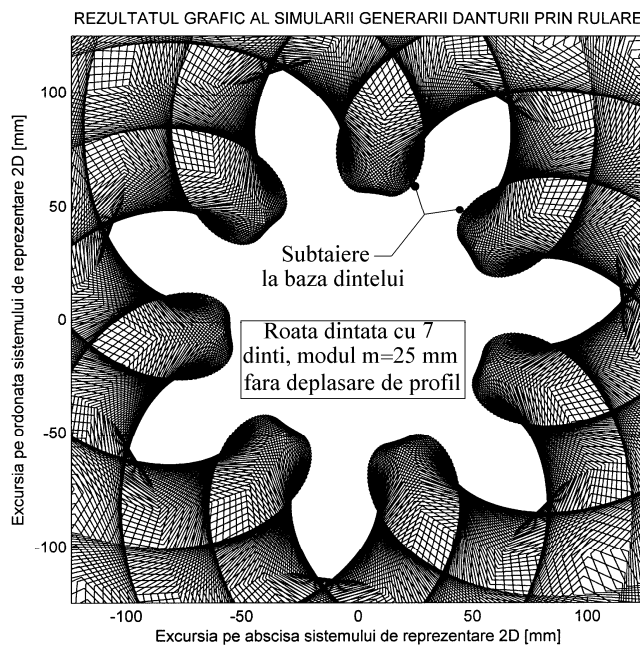


Figura 10. Ilustrarea fenomenului de subțiere în cazul generării unei roți dințate cu șapte dinți, fără deplasare de profil.

număr de dinți inferior lui 17. Un exemplu de generare a danturii unei roți dințate cu numai 7 dinți (modul $m=25\text{mm}$) se prezintă în figura 10. Fenomenul de subtăiere este aici și mai pronunțat.

Cu titlu informativ determinarea exactă a profilului roții dințate în aceste condiții (cu utilizarea programului DETPROF în care se prescrie $z=7$) nu mai este posibilă. Invităm cititorul la un exercițiu de observare și eventual corecție a problemei.

Utilizarea unei asemenea roți dințate implică obligatoriu eliminarea subtăierii în maniera cunoscută, prin deplasare de profil, cu simularea efectului acesteia prezentat în figura 11.

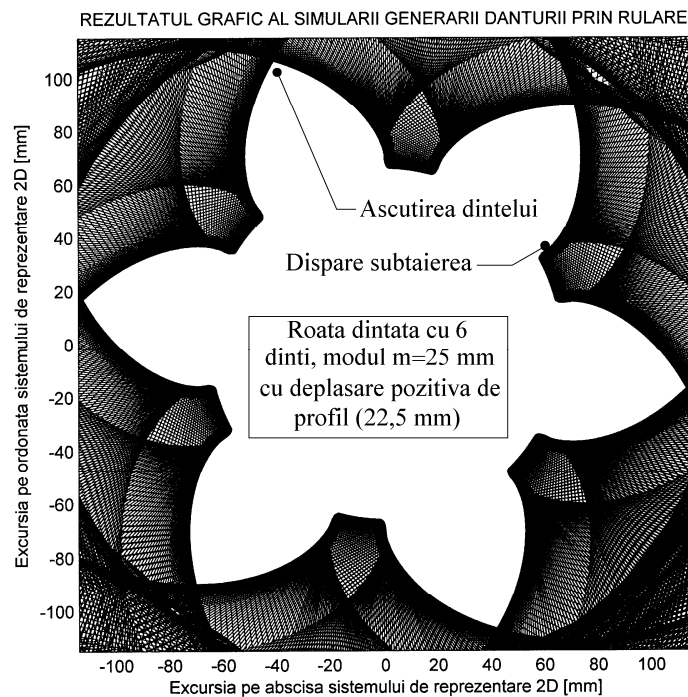


Figura 11. Ilustrare a eliminării fenomenului de subtăiere prin deplasare pozitivă de profil.

Generarea figurii s-a făcut cu ajutorul programului SIMGENDANT în care s-a prescrie $z=6$. Se poate utiliza la fel de bine și programul DETPROF, cu specificarea faptului că aici este posibilă determinarea profilului exact al roții. Pentru eliminarea subtăierii s-a

impus utilizarea unei deplasări de profil considerabile (aici 0,9·m, adică 22,5 mm). Fenomenul de ascuțire la vârful dintelui se manifestă cu intensitate maximă. Roata dințată astfel generată nu poate fi utilizată în aceste condiții. Există riscul strivirii-ruperii vârfului dintelui cu funcționarea improprie a angrenajului. Practica a impus o a doua soluție de compromis, utilizarea scurtării dintelui, adică a unui coeficient de cap al danturii mai mic decât cel standard (1,1·m pe figurile 4 b și 4 c). Cea mai simplă soluție de scurtare a dintelui se poate aplica la danturare prin utilizarea unui semifabricat cu diametru exterior mai mic.

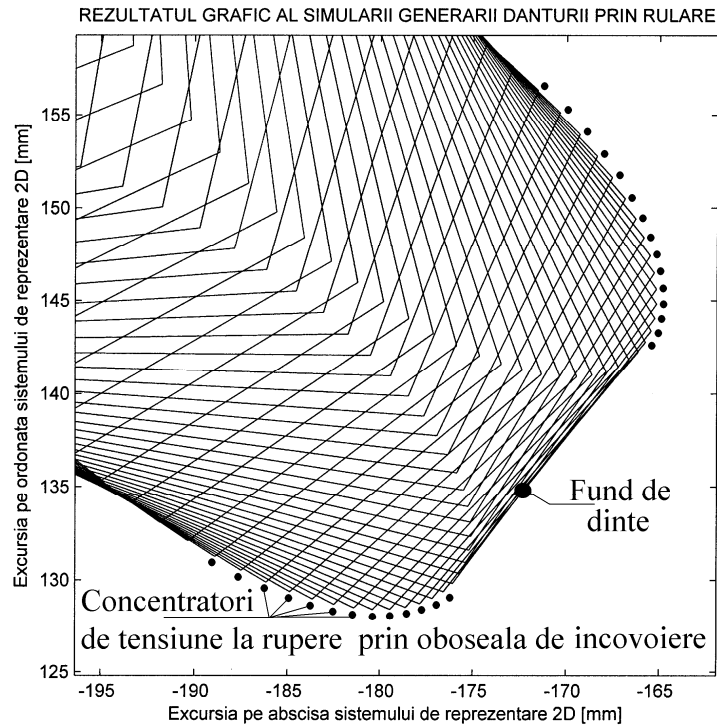


Figura 12. *Reliefaarea apariției concentratorilor de tensiune la baza dintelui.*

La generarea prin rulare cu sculă asociată cremalierii generatoare flancurile dinților roții dințate sunt definite în fapt printr-o succesiune de segmente tangente la profilul evolventic teoretic. Cu cât incrementul unghiului θ (conform figurii 4) este mai mic cu atât precizia de definire a profilului crește (crește numărul segmentelor de

tangență și scade lungimea acestora). În practică dezideratul este dificil de realizat (la prelucrarea prin mortezare ar presupune mișcări de rulare cu viteză extrem de mică și o mișcare principală rectilinie alternativă a sculei cu frecvență cât mai mare). Din acest motiv în practică se utilizează procedee de finisare a danturii, șeveruire, rodare respectiv rectificare. Rectificarea elimină și un alt neajuns al procedului de generare a danturii. Conform figurii 12 în zona de racordare dintre flancul și fundul dintelui se generează de către vârful sculei discontinuități de profil cu apariția de concentratori de tensiune care pot iniția ruperea prin oboseala de încovoiere.

O altă abordare în modelare permite simularea generării danturii prin considerarea ruletelor adică a traiectoriilor generate de toate punctele cremalierii generatoare în mișcarea de rulare (a curbelor cicloidală). Este evident că aceste rulete înfășoară profilul roții dințate. Utilizarea programului SIMGENDANTCURBE1 listat în continuare (și reluat în Anexa 4) permite reliefarea acestui aspect. Sunt extrapolate elementele de logică programată deja utilizate în programul SIMGENDANT. Se are în vedere generarea prin rulare a unei roți dințate cu caracteristici (modul, număr dinți) identice cu cea avută în vedere la simulările din figurile 5, 6, 7 și 8. Diferența principială față de programul SIMGENDANT constă din faptul că se definesc prin calcul puncte suplimentare pe fiecare tăiș al cremalierii (în afară de cele ce definesc capetele de segment). Aceste puncte au proiecția pe abscisă echidistantă (aici 1 mm). Folosind ecuațiile (12) și (13) se calculează și se memorează coordonatele pozițiilor succesive ale acestor puncte pentru rularea completă a cremalierii generatoare.

PROGRAM SIMGENDANTCURBE1

```
clear all;close all;
m=25;z=20;a=120;fi=90*pi/180;
alfa=20*pi/180;c=2.25*m*tan(alfa);b=0.5*m*(pi-
4.5*tan(alfa));dep=0*m;rr=0.5*m*(z);
xa(1)=a;xa(2)=a+b;xa(3)=a+b+c;xa(4)=xa(3)+b;xa(5)=xa(4)+c
;xa(6)=xa(5)+b;
ya(1)=1.1*m+dep;ya(2)=1.1*m+dep;ya(3)=-1.25*m+dep;ya(4)=-
1.25*m+dep;ya(5)=1.1*m+dep;ya(6)=1.1*m+dep;
```

```

g=1;
x1=xa(1);y1=ya(1);x2=xa(2);y2=ya(2);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;i=1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;end

x1=xa(2);y1=ya(2);x2=xa(3);y2=ya(3);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;end

x1=xa(3);y1=ya(3);x2=xa(4);y2=ya(4);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;end

x1=xa(4);y1=ya(4);x2=xa(5);y2=ya(5);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;end

x1=xa(5);y1=ya(5);x2=xa(6);y2=ya(6);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;end
npuncte=i-1;

for j=1:z+5;
    for i=1:npuncte;
        x(j*npuncte+i)=x(i)+j*m*pi;y(j*npuncte+i)=y(i);
    end
end

u1=size(x);u=u1(2);
xc=0;yc=0;
liminf_teta=0;limsup_teta=1000*pi/180;inc=360/(10*z)*pi/180;
;k=1;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:u
        x1a(k,i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        y1a(k,i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
        end;k=k+1;
    end
plot(x1a,y1a,'k');axis equal
axis([- (rr+1.6*m), (rr+1.6*m), - (rr+1.6*m), (rr+1.6*m) ])

```

```
xlabel('Abcisa reprezentarii [mm]');ylabel('Ordonata reprezentarii [mm]')
```

Reprezentarea grafică a acestor poziții succesive ale punctelor (folosind segmente de dreaptă între oricare două puncte succesive) și un zoom grafic în zona unui dinte permite obținerea reprezentării grafice din figura 13. Din considerente legate de estetica reprezentării

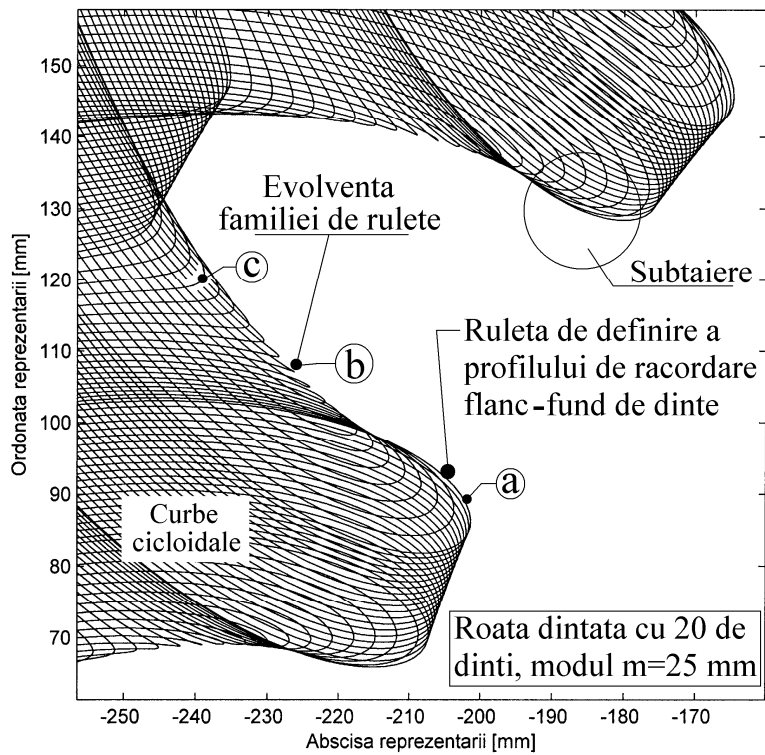


Figura 13. Detaliu de definierea a profilului roții dințate ca înfășurătoare a ruletelor generate de puncte distincte ale cremalierii generatoare.

grafice, incrementul unghiular de variație a unghiului θ folosit în programul SIMGENDANTCURBE1 este semnificativ mai mic în comparație cu valorile utilizate anterior.

Pe figura 13 apare evident faptul că profilul de roată dințată este delimitat de către ruletele cremalierii generatoare. Toate ruletele (curbele cicloidale) au aceleași ecuații parametrice, diferențele de alură apar datorită valorilor diferite ale ordonatelor $y(P_i)$ din ecuațiile (12) și (13). Punctele cu ordonată negativă (plasate sub rulantă) generează rulete caracterizate de existența unui lob de întoarcere (de exemplu curba **a** pe figura 13). Punctele cu ordonată zero, (plasate pe rulantă) generează rulete caracterizate de existența unui punct de întoarcere (de exemplu curba **b** pe figura 13). Conform elementelor teoretice privind generarea evolventei, curbele cu punct de întoarcere sunt chiar evolente. Punctele cu ordonată pozitivă (plasate deasupra rulantei) generează rulete caracterizate prin arc de întoarcere (de exemplu curba **c** pe figura 13).

Curba **a** definește racordarea flanc-fund de dinte. Ținând seama de alura acesteia se pot imagina și implementa tehnici de analiză asistată de calculator care să identifice automat existența subtăierii la baza dintelui și verificarea eventualei interferențe a roților dințate.

3. Aspecte teoretice ale simulării generării prin rulare ale conturilor poligonale închise (rulanta dreaptă, baza cerc).

Stadiul atins până în prezent permite ilustrarea prin modelare și simulare a elementelor anticipate în § 2.2. S-a văzut în finalul § 2.1. la nivelul figurii 3 că rularea unei figuri plane **Csr** (aici triunghi echilateral) solidară cu rulanta cerc, pe o curba plană fixă (bază, aici o dreaptă) generează ca înfășurătoare a pozițiilor succesive a **Csr** o curbă plană solidară cu baza, **Cgr**. Este absolut logic ca, plasându-ne în contextul § 3 și al figurii 4, să tratăm acum problema inversă. Rularea curbei **Cgr** solidară cu o rulantă dreaptă (care anterior a fost bază) pe un cerc fix utilizat ca bază (care anterior a fost rulantă) ar trebui să genereze curba **Csr** (ca înfășurătoare a pozițiilor succesive ale curbei

Cgr) pe baza utilizării relațiilor (12) și (13). Utilizarea programului TRIUNGHI 1 permite verificarea prin simulare a acestei concluzii.

Program TRIUNGHI1

```

close all;clear all
r=10;a=100;b=0;x(1)=0;y(1)=10;x(2)=8.660254;y(2)=-
5;x(3)=-8.660254;y(3)=-5;
x(4)=0;y(4)=10;k=4;l=1;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);y1(l)=b-
x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;if
x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
•for i=10:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);if i<x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else end
        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else end
            end;l=l+1;
    if miny<minminy;minminy=miny;else end
end
dim1=size(ord);dim=dim1(2);
%plot(abs,ord,'r','LineWidth',1.5);axis equal
L0/ord=minminy-ord;plot(abs,ord,'k','LineWidth',1.5)

```

```

L1/clear x1;clear y1;clear x;clear y;
L2/x=abs+5;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
L3/liminf teta=0;limsup teta=3*pi;inc=2*pi/100;
L4/for teta=liminf teta:inc:limsup teta;
    L5/for i=1:dim
        L6/x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        L7/y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    L8/end
    L9/plot(x1,y1,'k');hold on;axis equal
L10/end
L11/axis([(xc-1.2*r),(xc+1.2*r),(yc-1.2*r),(yc+1.2*r)])

```

Cea mai facilă opțiune este evident utilizarea demersului teoretic deja efectuat în orizontul figurii 4. O primă parte a programului (nemarcată cu bold) realizează sarcini foarte asemănătoare cu programul TRIUNGHI. Se generează o curbă **Cgr** suficient de lungă, provenită din rularea **Csr** cu mai mult de o rotație. Se reține din aceasta o zonă acoperitoare unei rotații, excluzând zonele de capăt de tip segment de dreaptă. Se definește o curbă **Cgr'** imagine în oglindă pentru **Cgr** (cu prima instrucțiune din L0/). Curba **Cgr'** are punctele de maxim cu ordonată nulă.

Rularea curbei **Cgr'** este efectiv simulată cu ajutorul zonei de program marcată cu bold, emulată cu foarte puternice similarități cu ale programului SIMGENDANT. În L2/se transferă curba **Cgr'** în reperul cremalierii din figura 4 b. (**Cgr'** devine tăiș principal de sculă de mortezat, punctele de ordonată maximă ale acesteia aparțin rulantei) și se definesc o serie de constante utilizate în condițiile figurii 4 și a ecuațiilor (12) și (13). De exemplu raza de rulare **rr=r**, aceeași cu a rulantei din figura 3. În L3/ se definesc limitele și incrementul de variație a parametrului de rulare (unghiul θ).

Calculul efectiv al coordonatelor de descriere a pozițiilor sculei se face în ciclul dintre liniile L4/ și L10/. În liniile L6/ și L7/ se materializează ecuațiile (12) și (13). Instrucțiunea din L9/ realizează desenarea pozițiilor succesive ale sculei, instrucțiunea din

L9/ realizează limitarea spațiului de reprezentare grafică. Rezultatul rulării programului TRIUNGHI1 este prezentat în figura 14.

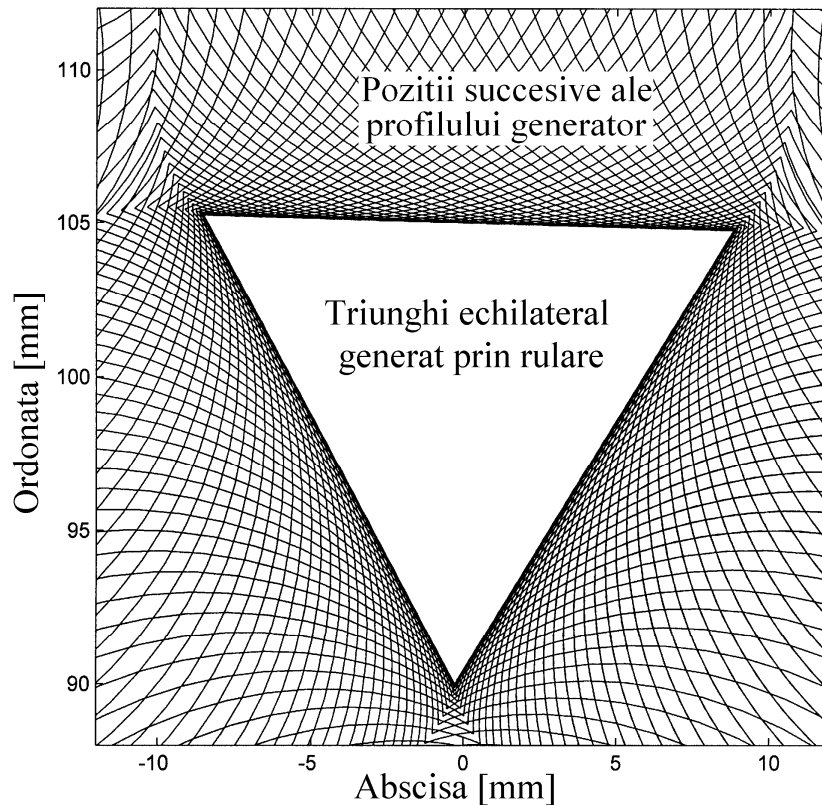


Figura 14. Reprezentare a rezultatului grafic al rulării programului TRIUNGHI1 (Anexa 5).

Apare evident faptul că s-a generat un triunghi echilateral ca înfășurătoare a pozițiilor succesive a curbei C_{gr}' (profilul generator).

Prin aceasta s-au probat practic prin modelare ambele demersuri teoretice, rularea cu rulantă cerc și baza dreaptă (conform cu § 2.1) și rularea cu rulantă dreaptă și baza cerc (conform cu § 3.1).

Evident că laturile triunghiului rezultă ca înfășurătoare a pozițiilor succesive ale curbei generatoare, profilul real este în fapt o succesiune de arce de curbă cu atât mai scurte cu cât incrementul parametrului θ este mai mic. Modificând constanta aditivă 5 a abscisei

$x=abs+5$ din linia L2/ se poate obține o poziție rotită a triunghiului echilateral generat în figura 14.

Se pot imagina acum o multitudine de scenarii de obținere a profilului generator și de verificare prin rulare a acestuia pentru orice figură plană. De exemplu în cazul unui pătrat cu latura de 10 mm se utilizează programul TRIUNGHI1 în care primele linii (subliniate și scrise înclinat) sunt substituite cu instrucțiunile:

```
close all;clear all;r=10*2^0.5;a=100;b=100;  
x(1)=-10;y(1)=-10;x(2)=10;y(2)=-10;x(3)=10;y(3)=10;  
x(4)=-10;y(4)=10;x(5)=-10;y(5)=-10;k=5;l=1;
```

iar în instrucțiunea marcată cu • variabila contoar i are limită inferioară -60.

Se obține astfel programul PATRAT1 (redat complet în Anexa 6).

Se definesc astfel pozițiile segmentelor ce definesc laturile

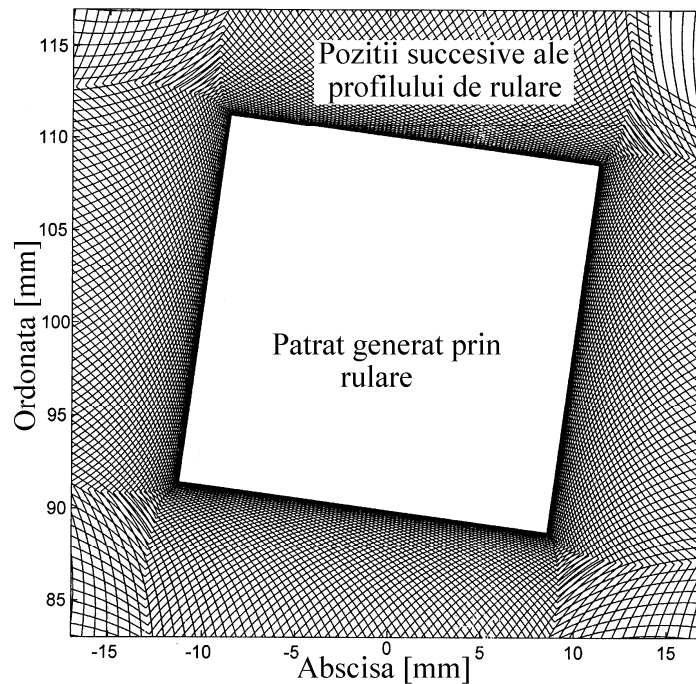


Figura 15. Rezultat grafic asupra modelării generării prin rulare a unui pătrat cu latura de 10 mm.

pătratului prin coordonatele vârfurilor acestuia.

Rularea programului în aceste condiții conduce la rezultatul grafic din figura 15. În linia L3/ s-a folosit un increment unghiular redus la jumătate, generat cu instrucțiunea: **inc=2*pi/200**. Și aici și sistematic ulterior se poate obține reprezentarea grafică a curbei **Cgr'** dacă se rulează separat instrucțiunea **plot(abs,ord,'k','LineWidth',1.5)** din instrucțiunea L0/. Descrierea sintetică a acestei curbe (de exemplu în vederea transferului ca tăiș pe sculă) este realizată cu ajutorul coordonatelor rezidente în matricile **abs** și **ord**. Se poate obține prin rulare un pentagon (programul PENTAGON1 din Anexa 7) sau un hexagon (programul HEXAGON1 din Anexa 8).

3.1 Limite ale generării prin rulare (rulanta dreaptă, baza cerc) a curbelor plane oarecare închise

Generarea prin rulare a curbelor plane are cel puțin o limitare importantă, cunoscută de practica în domeniu, (aparitia așa numitelor profile de trecere) reliefată aici, de exemplu, prin simularea generării unui poligon regulat de tip stea cu șase colțuri. Se utilizează același program TRIUNGHI1 în care primele instrucțiuni (subliniate) se substituie cu secvența:

```

close all;clear all
r=10;a=100;b=0;a1=0;a2=2.8868;a3=8.6603;
a4=5.7735;b1=10;b2=5;b3=0;
x(1)=a1;x(2)=a2;x(3)=a3;x(4)=a4;x(5)=a3;
x(6)=a2;x(7)=a1;x(8)=-a2;
x(9)=-a3;x(10)=-a4;x(11)=-a3;x(12)=-a2;x(13)=a1;
y(1)=b1;y(2)=b2;y(3)=b2;y(4)=b3;y(5)=-b2;y(6)=-b2;y(7)=-
b1;y(8)=-b2;
y(9)=-b2;y(10)=b3;y(11)=b2;y(12)=b2;y(13)=b1;
k=13;l=1;

```

Această secvență prescrie coordonatele celor 12 vârfuri ale poligonului în sistemul de coordonate x0y (conform cu figura 1).

Rularea numai a acestei secvențe de program împreună cu instrucțiunea **plot(x,y);axis equal** conduce la reprezentarea

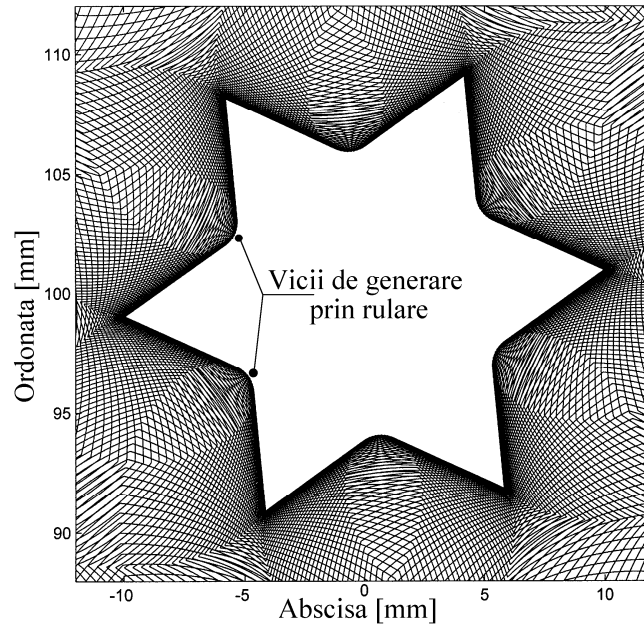


Figura 16. *Reliefarea unui viciu de generare prin rulare.*

acestui poligon (se definesc implicit laturile poligonului ca segmente

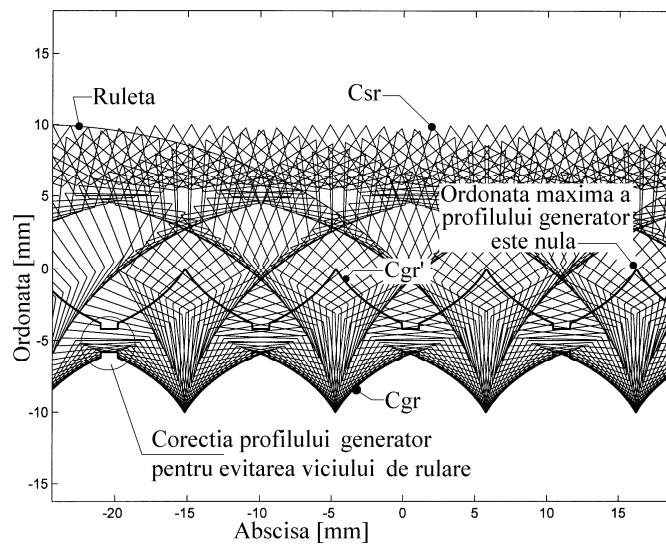


Figura 17. *Aspecte privind corecția profilului generator pentru evitarea viciului de generare prin rulare.*

între fiecare două vârfuri succesive).

Rularea programului TRIUNGHI1 astfel modificat (redat ca program fig16 în Anexa 9) conduce la rezultatul grafic din figura 16. Se observă un viciu relativ important de generare prin rulare, apariția unui arc de racordare (profil de trecere) în zona vârfurilor interioare, deși poligonul inițial din care s-a obținut prin dublă rulare figura 16 prezintă punct de intersecție între laturile ce definesc vârf interior. Apariția acestui arc de racordare este legată de faptul evident că pentru punctele profilului generator care nu se află pe rulantă, rularea se face cu viteză relativă nenulă. Același fenomen de apariție a unui profil de racordare a fost deja remarcat în cazul generării roților dințate (între flancul și fundul dintelui, figurile 7, 8, 9, 12 și 13). Din acest motiv angrenajul

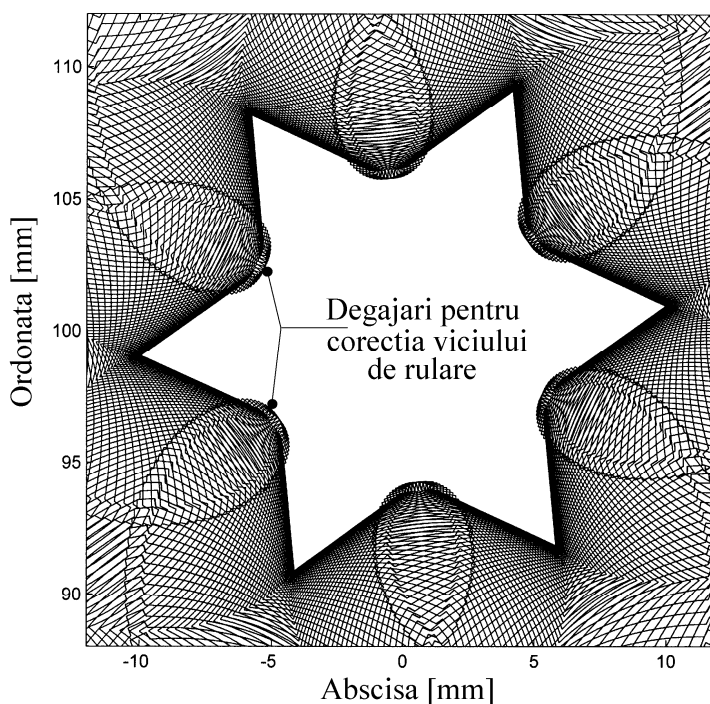


Figura 18. *Ilustarea efectului utilizării profilului generator corectat în înlăturare a viciului de rulare.*

utilizează un joc la fund adecvat, intrarea și ieșirea din angrenare se face prin contact al dinților în afara zonei de racordare.

Un exemplu de corecție în acest sens a profilului generator implicat în generarea figurii 16 este prezentat în figura 17. Se prezintă obținerea profilului **Cgr** în maniera cunoscută precum și corecția acestuia și a profilului generator **Cgr'**.

Se observă practicarea unei degajări pe ambele profile **Cgr** și **Cgr'** în zona implicată în generarea zonei cu viciu de rulare. Pentru generarea acesteia se inserează în program în zona imediat anterioară a ciclului de calcul a elementelor matricei **ord** a instrucțiunii: **prag=-6.1**; iar în interiorul ciclului a setului de instrucțiuni: **if ord(1)>prag;ord(1)=-5.8;else end**;

Rularea programului în aceste condiții conduce la obținerea curbei plane descrise în figura 18. Se observă apariția unor degajări în zona vârfurilor interioare. Acestea asigură plasarea punctului de intersecție a două laturi adiacente în afara corpului piesei delimitată exterior de profilul generat. Piesa conjugată (cu alezaj) se poate astfel monta.

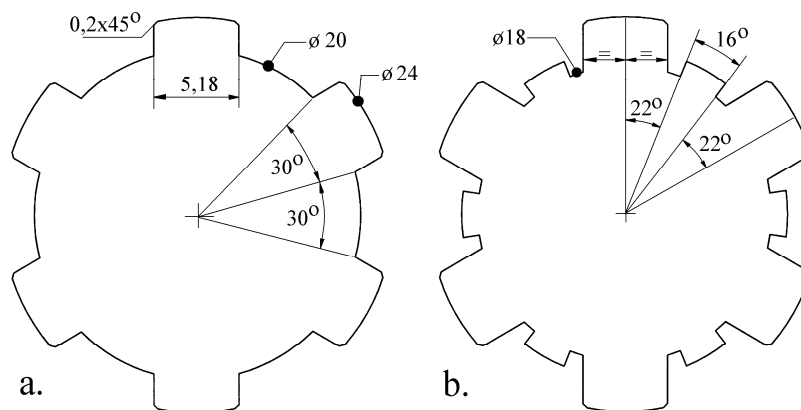


Figura 19. *Descriere principală a unui arbore canelat în secțiune transversală în două variante.*

Un compromis constructiv asemănător se utilizează cu mare eficacitate în corecția profilului generator (tăișul sculei) la prelucrarea prin rulare cu freză melc a arborilor canelați. Să exemplificăm acest aspect în cele ce urmează. Fie o vedere principală asupra unei secțiuni transversale printr-un arbore canelat, în două variante, conform figurii 19 a, b.

Să parcurgem acum cele două etape de modelare reliefate anterior pentru generarea secțiunii de arbore descris în figura 19 a. anume determinarea descrierii sintetice a profilului tăișului sculei de mortezat (prin rularea cerc pe dreaptă) și determinarea transpunerii acestuia prin rulare dreaptă pe cerc în vederea generării arborelui canelat.

Evident că va fi utilizat un program derivat din același program TRIUNGHI1, cu unele elemente de specific conform *listing*-ului:

```

Program CANELURAM1
close all;clear all;r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=9.6593;x(2)=-2.5882;y(2)=11.5176;
x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;y(15)=11.5176;
x(16)=2.5882;y(16)=9.6593;k=17;incdef=5.2871*cons/10;k=17
for i=73.2871*cons:-incdef:68*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-incdef:52*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-incdef:46.7129*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
u=60*cons;for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=120*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=180*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=240*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=300*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end
x(321)=x(1);y(321)=y(1);
plot(x,y);axis equal;k=321;l=1;

for i=0:2*pi/500:5/2*pi;u=-i;for j=1:k;
x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;end;end;l=l-1;
plot(x1,y1,'b');hold on;axis equal;maxx=-100;minx=100;
for i=1:l;if x1(i)>maxx;maxx=x1(i);else end
if x1(i)<minx;minx=x1(i);else end
x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;

```

```

k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc2=(maxx-minx)/4000;l=1;minminy=10000;
%for i=minx+0.001:inc2:maxx-inc2;miny=1000;ics=0;
    for i=10:inc2:90;miny=1000;abs(l)=i;
        for j=1:k-1;
            if i>x1(j);if i<x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else end
            if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;else end
end
diml=size(ord);dim=diml(2);
plot(abs,ord,'r','LineWidth',1.5);axis equal
ord=minminy-ord;
close all;plot(abs,ord,'k','LineWidth',1.5);axis equal;
hold on
clear x1;clear y1;clear x;clear y;x=abs+13;y=ord;
fi=pi/2;
xc=0;yc=100;
liminf_teta=0;limsup_teta=3*pi;inc=2*pi/500;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:dim
        x1(i)=xc+r*cos(fi-teta)+(x(i)-r*teta)*cos(fi-teta-
pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(i)=yc+r*sin(fi-teta)+(x(i)-r*teta)*sin(fi-teta-
pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-1.2*r),(yc+1.2*r)])

```

Diferențele esențiale față de programul TRIUNGHI1 apar conform zonelor marcate cu bold. Prima zonă (subliniată și marcată) realizează definirea punct cu punct a secțiunii transversale a canelurii din figura 19. a (curba **Csr** din sistemul **xOy** conform figurii 1). Execuția acesteia ca program distinct duce la obținerea grafică a acestei secțiuni.

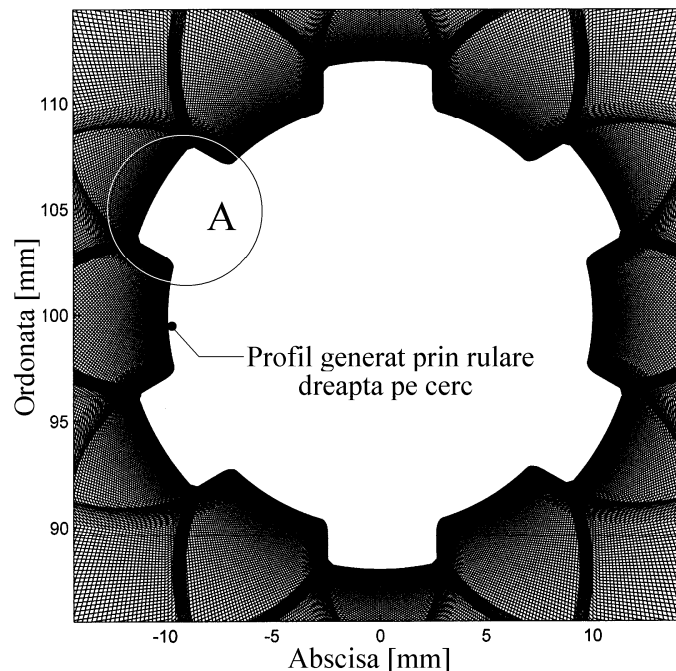


Figura 20. *Iustrativă privind generarea prin rulare a arborelui canelat cu secțiunea din figura 19 a. Simulare pe baza programului CANELURAM1.*

Celelalte diferențe marcate cu bold realizează prescrierea valorilor incrementilor folosiți la iterație. Cititorul interesat în experimentarea rulării acestui program va utiliza evident ca punct de plecare programul TRIUNGHI1 completat cu elementele de specific marcate.

Rezultatul generat prin simulare grafică cu ajutorul programului CANELURAM1 în aceste circumstanțe este prezentat în figura 20. Un detaliu grafic în zona A a figurii 20 este redat în figura 21. Din ambele figuri rezultă cu claritate generarea corectă a arcelor de picior și cap, a teșirilor și parțial corect a flancurilor rectilunii. Ca și anterior se observă și aici apariția viciului de rulare prin generarea unei curbe de racordare (profil de trecere) între flancul și fundul canelurii.

Ca și în cazul figurii 16, rezolvarea acestui inconvenient se poate face prin modificarea profilului generator (soluția prezentată în figura 17), cu efect în eliminarea profilului de trecere printr-o degajare (conform exemplului prezentat în figura 18).

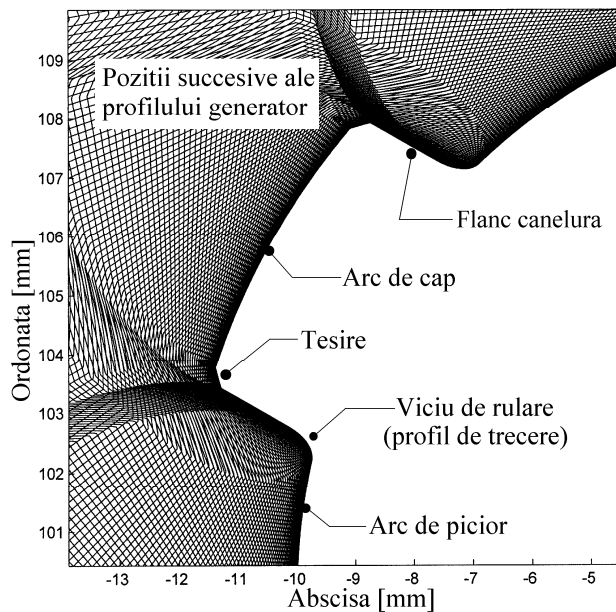


Figura 21. Detaliu grafic la figura 20 (zona A)

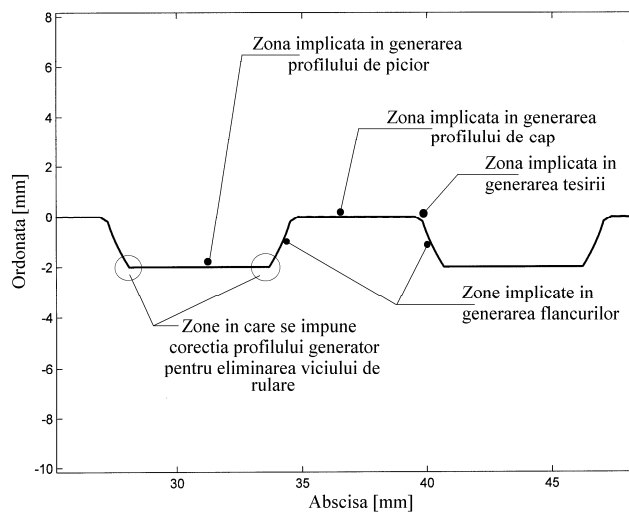


Figura 22. Detaliu grafic asupra profilului generator utilizat în figura 20 (obținut prin rulare cerc pe dreaptă).

Un detaliu grafic asupra profilului generator utilizat în generarea

canelurii din figura 20 este prezentat în figura 22. Acesta se poate obține rulând parțial programul CANELURAM1 până la instrucțiunea: ***close all;plot(abs,ord,'k','LineWidth',1.5);axis equal;*** marcată cu bold și italic în listing. Pe figură sunt evidențiate zonele implicate în definirea fiecărei porțiuni de pe profilul generat, precum și plasarea vârfurilor care generează profilele de trecere aferente viciilor de rulare.

Există însă o soluție mai simplă și mai eficientă de obținere a profilului generator care să permită realizarea degajării de subtăiere fără diminuarea lungimii flancurilor implicate în portanța torsională a canelurii. Să considerăm a doua variantă de arbore canelat, cu secțiunea transversală descrisă în figura 19 b. Aceasta păstrează toate caracteristicile anterioare ale secțiunii din figura 19 a. însă are prevăzut în dreptul fiecărui flanc o degajare ce acoperă un unghi de cca. $5,2^\circ$.

Să considerăm acum această secțiune de arbore canelat ca definind profilul generator (rulare cerc pe dreaptă) și apoi să utilizăm acest profil în generarea arborelui canelat (rulare dreaptă pe cerc). Se poate utiliza același program CANELURAM1 dar cu substituirea zonei de debut (marcată cu bold și subliniată) cu setul de instrucțiuni:

```
close all;clear all;r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-incdef:68*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-incdef:52*cons;x(k)=10*cos(i);
y(k)=10*sin(i);
k=k+1;end;
incdef=5.2871*cons/10;
for i=52*cons:-incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
    y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
```

```

end;u=120*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
    y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
    x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
    x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
    x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis equal
k=321;l=1;

```

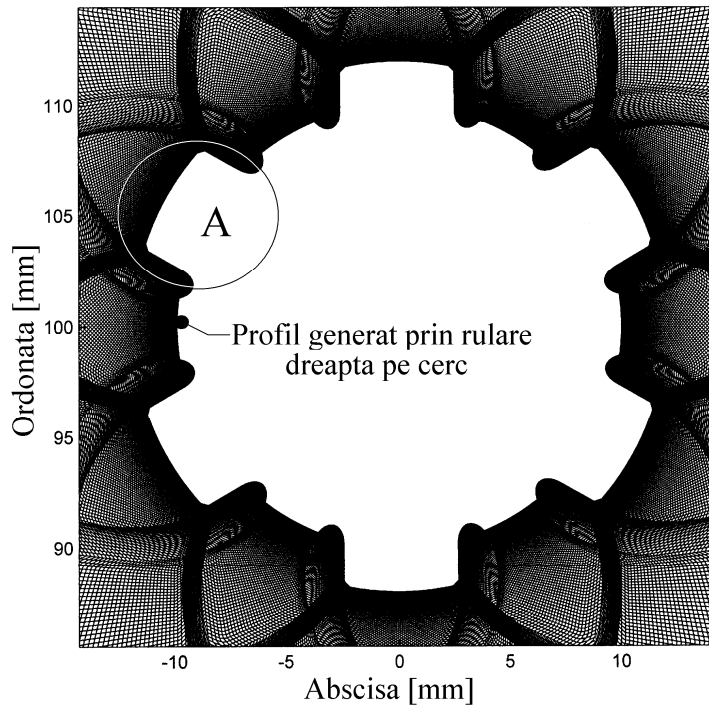


Figura 23. *Iustrativă privind generarea prin rulare a arborelui canelat cu profil de trecere deplasat. Simulare pe baza programului CANELURAM2 (listing complet redat în Anexa 12).*

Execuția numai a acestei secvențe ca program distinct duce la obținerea reprezentării grafice a secțiunii de arbore canelat din figura 19 b.

Rularea programului în aceste condiții (redat sub numele CANELURAM2 în Anexa 12) conduce la obținerea reprezentării din figura 23. Se observă apariția unor degajări care deplasează viciul de rulare (profilul de trecere) sub punctul de intersecție dintre flancuri și arc de fund al canelurii. Aspectul este mai evident în figura 24 în care

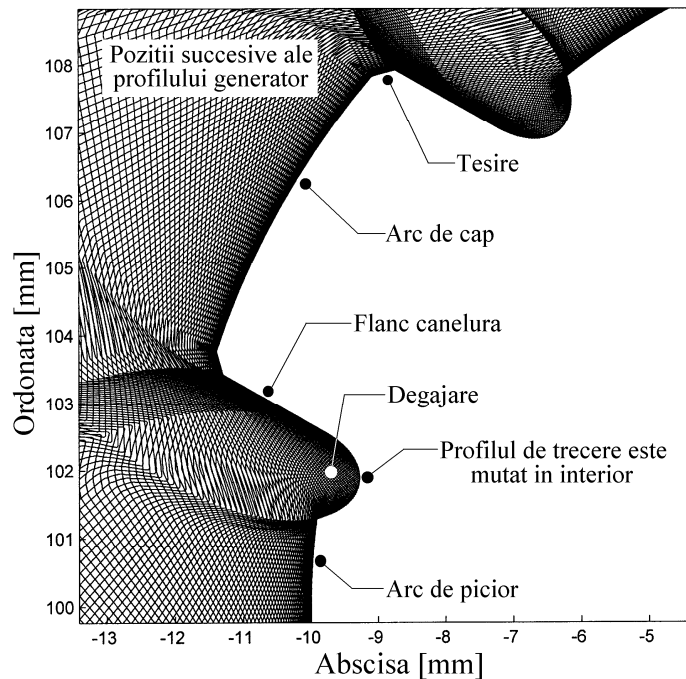


Figura 24. Detaliu grafic la figura 23 (zona A)

se prezintă o vedere de detaliu la figura 23 (în zona A). Prin simulare se poate decide asupra adâncimii necesare a degajării care să asigure profil rectiliniu al flancurilor deasupra cercului ce conține arcele de picior ale canelurii. În acest fel montarea contrapiesei pe arbore (de tip bucșă canelată) devine foarte facilă. Se asigură în același timp păstrarea capacității portante a canelurii, mărimea suprafeței de contact pe flancuri nefiind afectată.

Evident că programul CANELURAM2 permite și obținerea descrierii sintetice a profilului generator (regăsit ca tăiș pe sculă la obținerea secțiunii canelate prin rulare dreaptă pe cerc). Coordonatele

acestui profil se găsesc în matricile **abs** (abscisa) și **ord** (ordonata). Imaginea grafică parțială a acestui profil se prezintă în figura 25. Aceasta s-a obținut prin rularea programului până la instrucțiunea

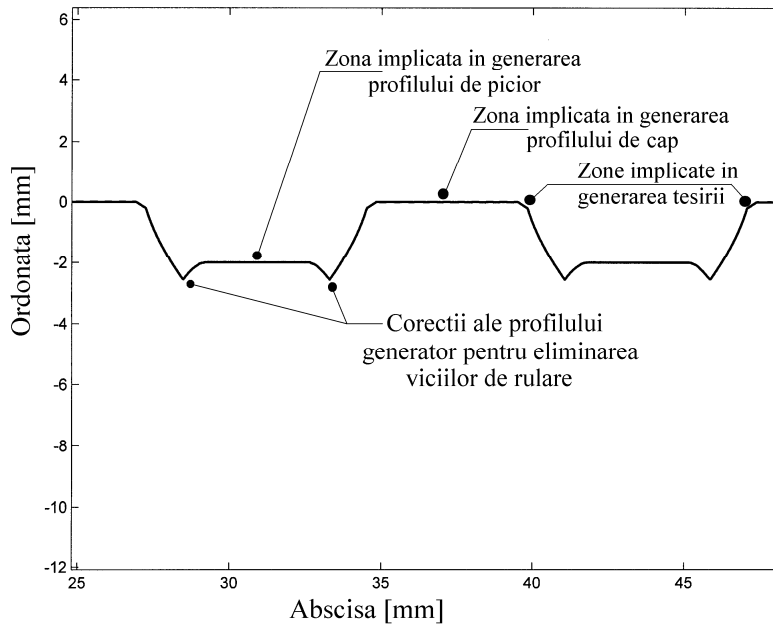


Figura 25. Detaliu grafic asupra profilului generator utilizat în figura 23 (obținut prin rulare cerc pe dreaptă).

`close all;plot(abs,ord,'k','LineWidth',1.5);axis equal;` (în program marcată cu bold italic) inclusiv.

În comparație cu profilul generat în figura 22, aici se observă clar dezvoltarea unor vârfuri de corecție implicate în generarea degajărilor evidențiate pe figura 24. Soluția de corecție a viciului de rulare este mult mai simplă și se generează automat.

În concluzie, ca notă generală, sinteza profilului generator se poate face sintetic prin determinarea coordonatelor punct cu punct urmare a rulării simulate cerc pe dreaptă. Există și varianta practică, prin utilizarea unui cuțit roată de mortezat, cu caracteristicile exacte ale curbei plane închise de generat. Prin rularea cu rulanta cerc și baza o dreaptă se realizează profilul **Cgr** care rotit cu 180° se constituie în

profil generator C_{gr} regăsit pe scula pieptene (liniară) de mortezat. Profilul generator atașat unei rulante rectilinii pe bază circulară se transpune prin rulare pe piesa finită.

4. Elemente de *feed-back*. Determinarea profilului cuțitului pieptene de mortezat roți dințate cu profil evolventic

În § 3.1 s-a realizat modelarea definirii prin rulare a roților dințate (rulanta o dreaptă și baza un cerc) considerându-se un profil generator predefinit, de tip cremalieră generatoare. Este absolut logic să se realizeze în continuare o verificare de principiu care să răspundă următoarei întrebări: poate fi definită grafic cremaliera generatoare în termenii capitolului 2, ca fiind generată prin rulare cu rulanta cercul de rostogolire al roții și baza o dreaptă fixă? Evident că aici curba C_{sr} este roata dințată (al cărei profil este definit de exemplu conform figurii 6) solidară cu rulanta. Un răspuns pozitiv la această întrebare ar confirma suplimentar corectitudinea demersului de modelare și simulare efectuat aici.

În acest sens s-a realizat sinteza unui program numit DETPROFA care realizează compilația (practic alipirea) a două programe deja prezentate anterior, anume a programului SIMGENDANT și a programului TRIUNGHI, cu *listing*-ul prezentat aici (și în Anexa 13):

Program DETPROFA

```
clear all;close
all;m=25;z=20;a=120;fi=90*pi/180;m1=m;
alfa=20*pi/180;c=2.35*m*tan(alfa);b=0.5*m*(pi-
4.7*tan(alfa));dep=0*m;
rr=0.5*m*(z);
x(1)=a;x(2)=a+b;x(3)=a+b+c;x(4)=x(3)+b;x(5)=x(4)+c;x
(6)=x(5)+b;
```

```

y(1)=1.1*m+dep;y(2)=1.1*m+dep;y(3)=-
1.25*m+dep;y(4)=-
1.25*m+dep;y(5)=1.1*m+dep;y(6)=1.1*m+dep;ndinti=z+5;
for j=1:ndinti;for
i=1:6;x(j*6+i)=x(i)+j*m*pi;y(j*6+i)=y(i);end;end
xc=0;yc=0;l=1;liminf_teta=0;limsup_teta=2.5*pi;inc=2*pi
i/300+0.00001;
for teta=liminf_teta:inc:lmsup_teta;
    for i=1:6*(ndinti)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);l=l+1;
    end
end;l=l-1;%plot(x1,y1,'k');hold on;axis equal
axis([- (rr+1.4*m1), (rr+1.4*m1), -
(rr+1.4*m1), (rr+1.4*m1)])
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc=2*pi/2000;l=1;
for i=0:inc:2.1*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);yb=ya+5*sin(i);dif=xb-
xa;if dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);if
dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-
ni)/dif1;inty=intx*m(j)+n(j);dist=(abs(intx-
xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;

```

```

        if intx==x1(j+1);if inty==y1(j+1);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
        if intx==x1(j);if inty==y1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
        if intx<x1(j+1);if intx>x1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
        if intx>x1(j+1);if intx<x1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;mindist=dist;e
lse end;else end;else end;
    else end
end;l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);
clear x;clear y;x=absc;y=ord;clear absc;clear
ord;clear x1;clear y1;
dim1=size(x);k=dim1(2);l=1;r=rr
for i=0:2*pi/100:2*pi;u=i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)-
y(j)*sin(u);y1(l)=b+x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end;l=l-1;
plot(x1,y1,'k');hold on;axis equal;maxx=-
100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;
k=1;

```

```

for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;n(k)=y1(i)-
m(k)*x1(i);k=k+1;
end; k=k-1;
inc2=(maxx-minx)/3000;l=1;minminy=10000;

```

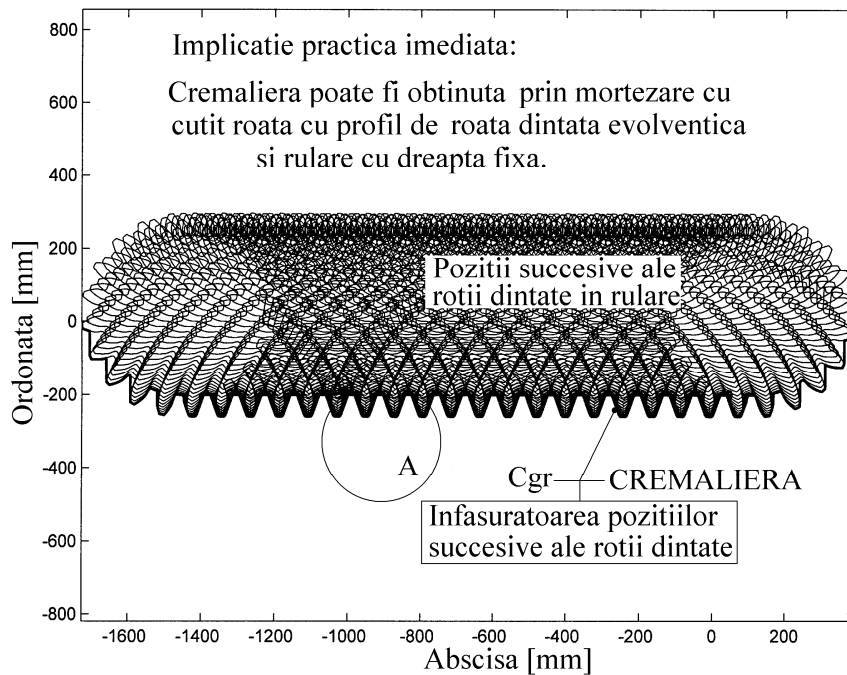


Figura 26. Rezultatul grafic al execuției programului DETPROFA.

```

for i=minx+0.001:inc2:maxx-
inc2;miny=1000;ics=0;abs(l)=i;
    for j=1:k-1;
        if i>x1(j);if i<x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else
end
        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else end;else
end
    end;l=l+1;

```

```

if miny<minminy;minminy=miny;else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'k','LineWidth',1.5);axis equal

```

Rezultatul rulării programului DETPROFA este prezentat în figura 26. Apare evident aici și mai ales în figura 27 (un detaliu din zona A a figurii 26) faptul că înfășurătoarea pozițiilor succesive ale roții dințate este o cremaliera generatoare. Dacă în § 3.1 un segment rectiliniu de pe cremaliera generatoare producea prin rulare un arc de evolventă, aici un arc de evolventă de pe roată generează prin rulare un segment de dreaptă pe cremaliera. Implicația imediată -cunoscută în practică- este legată de ilustrarea faptului că este posibilă generarea prin rulare a unei cremaliere generatoare (de exemplu, conform figurii 19, prin procedeul

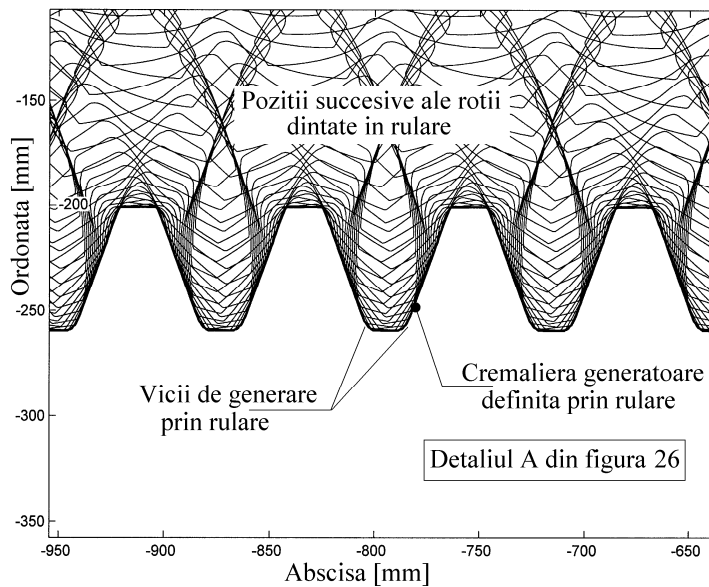


Figura 27. Detaliu grafic al figurii 26. Definierea prin rulare a cremalierei generatoare.

cu dreapta fixă) cu utilizarea unei scule de tip cuțit roată de mortezat. O imagine mai completă a cremalierei generatoare obținute prin rulare în condițiile figurii 26 se prezintă în figura 27 (zoom-in realizat prin

intermediul facilităților de explorare grafică Matlab).

Vârful dinților cremalierii este corect și complet definit, la baza acestora însă apare arc de racordare ca viciu de rulare, în condiții asemănătoare cu cele din § 3.1. În practică, la utilizarea cremalierii transferate pe sculă pentru danturare (procedeul cu cercul fix), implicațiile acestui neajuns sunt eliminate prin faptul că fundul cremalierii nu participă la așchiere, diametrul de cap al dintelui pe roata dințată rezultă din semifabricat. Scula așchietoare prelevează material numai din zona delimitată de flancuri și fundul dintelui.

5. Alte aplicații ale generării prin rulare dreaptă pe cerc

În afara aplicației consacrate (obținerea roților dințate cu profil de flanc evolventic) practica construcției organelor de mașini [2, 4] a generalizat și alte aplicații ale generării prin rulare dreaptă pe cerc. Unele dintre cele mai interesante se referă la obținerea roților utilizate în transmisia cu lanț, respectiv a roților de clichet.

5.1 Generarea prin rulare dreaptă pe cerc a roților transmisiilor cu lanț

Lanțul utilizat în transmisii mecanice este un organ de mașină rigid la tracțiune pe ramura activă și flexibil la înfășurare pe roți. Acesta permite transmiterea sub raport selectabil constructiv (prin intermediul numerelor de dinți ale roților) a mișcării de rotație, cu vehicularea unor momente de torsiune mari.

Spre deosebire de transmisiile cu curele netede (late sau trapezoidale, la care transmiterea mișcării se face prin frecare) transmisiile cu lanț realizează raport constant, fără patinare-alunecare. Lanțul este alcătuit din bolțuri și zale. Pe bolțuri se dispun bușe libere la rotație (ajustate cu joc). În funcționare bușele intră succesiv în locașuri practicate pe roți, transmisia făcându-se prin moment de

torsiune aplicat periferic. În funcționare, intrarea și ieșirea fiecărei bucșe în locașul corespunzător pe fiecare roată se face evident printr-un mecanism de rulare (dreaptă pe cerc). De aici concluzia că profilul roții de lanț trebuie să îndeplinească o serie de condiții după cum urmează:

1. Disponerea periodică echidistantă a bolțurilor pe lanț implică o dispunere periodică a locașurilor aferente bucșelor de bolț pe roată. Lungimea arcului de cerc dintre două locașuri succesive (între care se definește dintele roții), măsurat pe raza de definire a rulantei, este evident exprimabilă ca un multiplu al constantei π . Această lungime se regăsește obligatoriu între axele a fiecare două bolțuri succesive.

2. Locașurile aferente bucșelor de bolț pe roată trebuie să fie dispuse perfect echidistant unghiular. Pe arcul de contact dintre lanț și roată nu există mișcare relativă (patinare sau alunecare). O perfectă divizare a locașurilor asigură o implicare completă în transmiterea momentului de torsiune a tuturor dinților de pe arcul de contact, cu evitarea riscului de rupere-strivire.

3. Profilul de flanc al fiecărui dinte trebuie să asigure intrarea-ieșirea bucșelor în locașuri (încărcarea/descărcarea bolțurilor) fără a induce forțe de întindere suplimentare în lanț. Este de intuit faptul că pentru o transmisie cu lanț care funcționează pe ambele sensuri de rotație cele două flancuri ale fiecărui dinte trebuie să fie identice.

Îndeplinirea condițiilor enumerate anterior poate fi asigurată evident dacă profilul geometric din zona de contact al roților cu lanțul se generează prin rulare. Evident că se pot utiliza considerentele teoretice și de modelare asistată de calculator deja definite anterior în cazul rulării dreaptă pe cerc. Problema aparent dificilă aici este definirea profilului generator. Ținând seama însă de faptul că profilul roților poate fi privit ca înfășurătoare a pozițiilor succesive ale bucșelor de bolț este ușor de imaginat un profil generator alcătuit dintr-o succesiune de cercuri (cu diametrul bucșelor) plasate echidistant (la aceeași distanță cu cea dintre bolțuri).

Unele considerente de natură geometrică ale definirii unui astfel de profil generator sunt prezentate în figura 28. Sunt utilizate o succesiune de arce de cerc (cu diametrul bucșelor, este de intuit faptul

că în rulare nu se utilizează cercul în întregime) și de segmente de dreaptă. În contextul teoretic al figurilor 4 și 4 b., aferente rulării

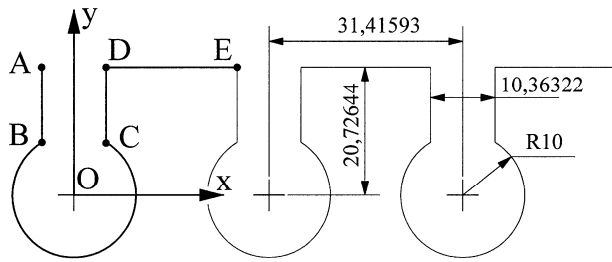


Figura 28. Exemplu de descriere grafică a profilului generator utilizat în rularea dreaptă pe cerc pentru simularea generării roților de lanț.

dreaptă pe cerc, sistemul xOy este plasat cu originea în centrul arcului de cerc al unui sector generic al profilului generator alcătuit din

conturul continuu ABCDE.

Imaginând

transpunerea pe un

proces real de prelucrare, acest contur ar corespunde tăișului de definire a unui dinte așchietor al unei scule de mortezat. Pe figura 28 s-au folosit elemente dimensionale cu caracter de particularizare (de exemplu distanța dintre axele a două bucșe succesive este de $10 \cdot \pi$). Rulanta este evident abscisa sistemului xOy . Pentru generarea prin rulare a unei roți de lanț cu număr oarecare de dinți se utilizează un profil generator cu lungime (număr de sectoare generice, sau număr de dinți) corepunzătoare obținută prin expandarea profilului ABCDE (în aceeași manieră cu procedura folosită la obținerea cremalierii generatoare, de exemplu în programul SIMGENDANT).

Elementele specifice obținerii prin rulare a roților de lanț sunt fixate prin simulare grafică cu ajutorul programului SIMURLAN cu listingul prezentat în continuare (și reluat în Anexa 14):

Program SIMURLAN

```
L1/clear all;close all;m=10;z=10;
L2/fi=90*pi/180;m1=m;rr=0.5*m*(z);dep=0*m;
L3/x(1)=-5.1816;y(1)=20.7264+dep;x(2)=-5.1816;
L4/y(2)=8.5528+dep;k=3;cons=pi/180;max=58.79102;
L5/min=-238.79102;inc=(max-min)/30;
L6/for i=min:inc:max;x(k)=10*cos(i*cons);
L7/y(k)=10*sin(i*cons)+dep;k=k+1;end;
```

```

L8/x(k)=5.1816;y(k)=8.5528+dep;x(k+1)=5.1816;
L9/y(k+1)=20.7266+dep;x(k+2)=26.2343;
L10/y(k+2)=20.7264+dep;k=k+2;
L11/ndinti=z+5;for j=1:ndinti;for i=1:k;
    L12/x(j*k+i)=x(i)+j*m*pi;y(j*k+i)=y(i);
L13/end;end;plot(x,y);axis equal;
L14/xc=0;yc=0;l=1;
L15/liminfteta=0;limsupteta=2.3*pi;
L16/inc=2*pi/100+0.00001;
L17/for teta=liminfteta:inc:limsupteta;
    L18/for i=1:k*(ndinti+1)
        L19/x1(l)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)
* cos(fi-teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        L20/y1(l)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)
* sin(fi-teta-pi/2)+y(i)*cos(fi-teta-pi/2);l=l+1;
    L21/end;end;l=l-1;
L22/plot(x1,y1,'k');hold on;axis equal
L23/axis([- (rr+1.6*m1), (rr+1.6*m1), -
(rr+1.6*m1), (rr+1.6*m1)])
L24/Title('REZULTATUL GRAFIC AL SIMULARII GENERARII
ROTII DE LANT PRIN RULARE')
L25/xlabel('Excursia pe abscisa sistemului de
reprezentare 2D [mm]')
L26/ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')
%rutina de determinare a profilului
L27/k=1;
L28/for i=1:l-1;
    L29/dif=(x1(i+1)-x1(i));
    L30/if dif==0;dif=0.00000000001;else end
    L31/m(k)=(y1(i+1)-y1(i))/dif;
    L32/n(k)=y1(i)-m(k)*x1(i);k=k+1;
L33/end;k=k-1;k
L34/inc=2*pi/10000;l=1;
L35/for i=0:inc:2.4*pi;mindist=1000;
    L36/xa=xc;ya=yc;xb=xa+5*cos(i);yb=ya+5*sin(i);
L37/dif=xb-xa;if dif==0;dif=0.00000000001;else end;
    L38/mi=(yb-ya)/dif;ni=ya-xa*mi;
    L39/for j=1:k-1; dif1=mi-m(j);
L40/if dif1==0;dif1=0.00000000001;else end;
    L41/intx=(n(j)-ni)/dif1;inty=intx*m(j)+n(j);
L42/dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;

```

```

L43/dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
L44/if dist1<dist;if intx==x1(j+1);
L45/if inty==y1(j+1);if dist<mindist;absc(l)=intx;
L46/ord(l)=inty; mindist=dist;
L47/else end;else end;else end;
L48/if intx==x1(j);if inty==y1(j);
L49/if dist<mindist; absc(l)=intx;ord(l)=inty;
L50/mindist=dist;else end;else end;else end;
L51/if intx<x1(j+1);if intx>x1(j);
L52/if dist<mindist;absc(l)=intx;ord(l)=inty;
L53/mindist=dist;else end;else end;else end;
L54/if intx>x1(j+1);if intx<x1(j);
L55/if dist<mindist;absc(l)=intx;ord(l)=inty;
L56/mindist=dist;else end;else end;else end;
L57/if inty<y1(j+1);if inty>y1(j);
L58/if dist<mindist;absc(l)=intx;ord(l)=inty;
L59/mindist=dist;else end;else end;else end;
L60/if inty>y1(j+1);if inty<y1(j);
L61/if dist<mindist;absc(l)=intx;ord(l)=inty;
L62/mindist=dist;else end;else end;else end;
L63/else end
L64/end;l=l+1;
L65/end
L65bis/%plot(absc,ord,'r','LineWidth',1.5);
%rutina pentru limitarea diametrului
L66/dim1=size(absc);dim=dim1(2);razlim=rrr+8;
L67/for i=1:dim
L68/r(i)=sqrt(abs(absc(i))^2+abs(ord(i))^2);
L69/if r(i)>=razlim;r(i)=razlim;else end
L70/if ord(i)==0;ord(i)=0.00000001;else end
L71/if absc(i)>0;
L72/if ord(i)>0;tetha(i)=atan(abs(ord(i)))/
abs(absc(i)); else end;else end
L73/if absc(i)<0;if ord(i)>0;tetha(i)=pi/2-
atan(abs(ord(i))/abs(absc(i)))+pi/2;
else end;;else end
L74/if absc(i)<0;if ord(i)<0;
tetha(i)=atan(abs(ord(i))/abs(absc(i)))+pi;else end;
else end
L75/if absc(i)>0;if ord(i)<0;
tetha(i)=pi/2-atan(abs(ord(i))/abs(absc(i)))+3*pi/2;
else end;else end

```

```

L76/end
L77/for i=1:dim;
L78/abscl(i)=r(i)*cos(tetha(i));
L79/ordl(i)=r(i)*sin(tetha(i));
L80/end
L81/plot(abscl,ordl,'b','LineWidth',1.5)

```

Programul SIMURLAN are foarte multe elemente comune cu programul SIMGENDANT respectiv DETPROF, motiv pentru care în prezentare se va insista pe elementele de natură specifică. Formal se folosește o constantă **m=10** (linia L1/), echivalentul modulului din programul SIMGENDANT, utilizată pentru exprimarea pasului lanțului (distanța dintre axele a două bolțuri succesive) respectiv a razei de rulare (raza bazei, linia L2/). Se consideră generarea unei roți cu **z=10** dinți.

Și aici se va folosi noțiunea de profil deplasat (variabila **dep**, deocamdată nulă, din linia L2/) pentru rațiuni ce vor rezulta ulterior. În liniile L3/ și L4/ se definesc coordonatele de capăt ale segmentului AB în condițiile din figura 28. Intre liniile L5/ și L7/ se realizează definirea coordonatelor punctelor de definire a arcului de cerc BC (coordonatele sunt rezidente în matricile **x** respectiv **y**). Se consideră că arcul (pentru care se precizează limitele unghiulare **min** și **max** de plasare a punctelor B și C) este definit prin 30 de puncte distincte, echidistante unghiular. Evident că descrierea fizică a acestuia se va face astfel prin intermediul a 29 de segmente (coarde). Numărul de puncte poate fi evident crescut sub rezerva creșterii timpului de calcul total la rularea programului. În liniile L8/, L9/, L10/ se prescriu valorile coordonatelor punctelor de definire a segmentelor CD respectiv DE. Variabila contoar **k** se generează automat în ciclul de calcul pe arcul de cerc.

În ciclul de instrucțiuni dintre liniile L11/ și L13/ se realizează definirea completă a profilului generator utilizat pentru definirea roții de lanț prin rulare dreaptă pe cerc. Se realizează de fapt replicarea, expandarea conturului ABCDE definit anterior. Executarea

numai a secvenței de program de până la linia L13/ inclusiv, permite reprezentarea grafică a acestui profil generator.

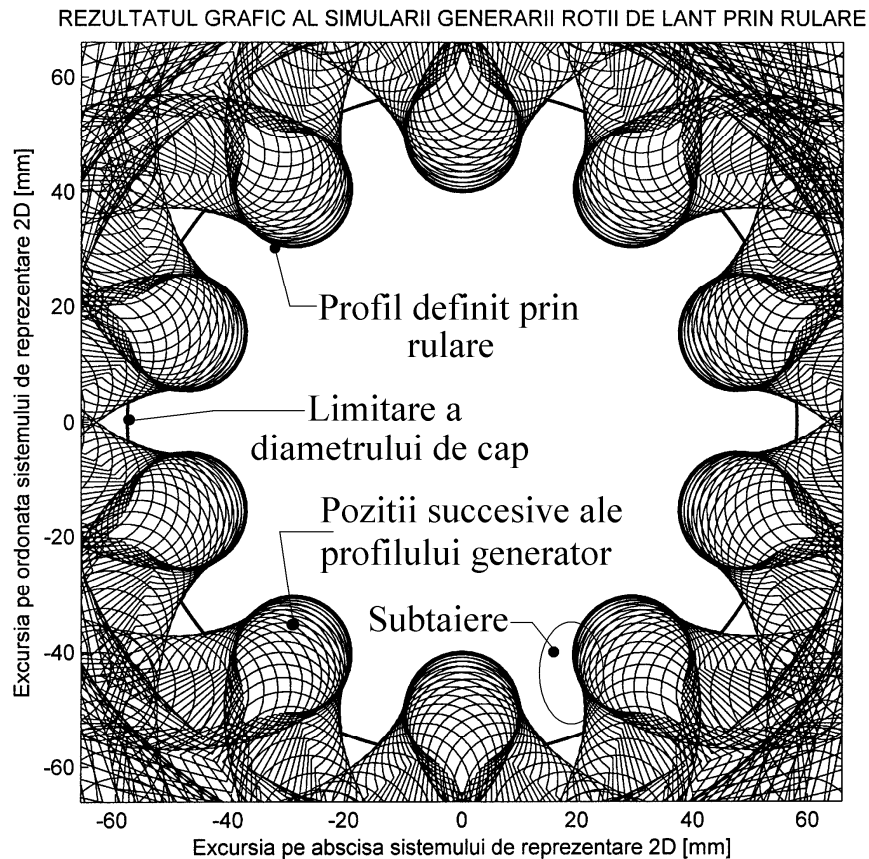


Figura 29. Rezultatul rulării programului SIMURLAN. Definierea profilului unei roți de lanț cu 10 dinți, cu limitarea diametrului de cap.

În ciclul de instrucțiuni dintre liniile L14/ și L21/ se realizează calculul coordonatelor punctelor de definire a profilului generator pentru toate pozițiile corespunzătoare rulării și definirii complete a roții de lanț. Pentru calculul coordonatelor (cu rezultatele rezidente în matricile $\mathbf{x1}$ și $\mathbf{y1}$) se utilizează elementele de modelare propuse deja în figura 4 și ecuațiile parametrice (12) și (13) ale traiectoriilor (cicloidelor sau ruletelor) punctelor de definire ale profilului generator. Cele două ecuații au fost preluate cu adaptări

minimale și utilizate în liniile L19/ și L20/. În linia L16/ se prescrie incrementul de variație a parametrului θ al ecuațiilor, cu instrucțiunea `inc=2*pi/100+0.00001`. Execuția secvenței de program dintre liniile L1/ și L23/ inclusiv, permite obținerea profilului grafic al roții de lanț ca înfășurătoare a pozițiilor succesive ale profilului generator conform figurii 29. Instrucțiunile din liniile L24/, L25/ și L26/ înscriu titlul, etichete pe abscisă și ordonată pe spațiul de reprezentare.

Cu ajutorul grupului de instrucțiuni dintre liniile L27/ și L65bis/ se realizează determinarea profilului roții de lanț, în exact aceeași manieră cu cea prezentată în programul DETPROF (secvența dintre liniile L1/ și L24/).

Cu ajutorul grupului de instrucțiuni dintre liniile L66/ și L81/ se realizează analiza profilului roții și se limitează raza de cap a acesteia la valoarea `razlim=rr+8` (altfel profilul rezultă cu ascuțire severă la vârful). Intre liniile L67/ și L76/ se realizează conversia descrierii profilului din coordonate ortogonale în coordonate polare și limitarea efectivă a razei de cap (instrucțiunea din L69/).

În ciclul de instrucțiuni de la L77/ la L81/ se realizează conversia descrierii profilului din coordonate polare în coordonate ortogonale și reprezentarea grafică a acestuia.

Rularea completă a programului SIMURLAN permite obținerea rezultatului grafic din figura 29. Spre deosebire de programele anterioare durata de execuție a programului este semnificativ mai mare în principal din cauza complexității definirii profilului generator (pentru care au fost utilizate aici un număr total de 576 puncte, respectiv 575 segmente distincte). Este și acesta un argument pentru utilizarea în simulare a unor calculatoare performante.

Pe figura 29 se observă clar maniera de definire a profilului de roată de lanț ca înfășurătoare a pozițiilor succesive ale profilului generator, descrierea profilului cu limitarea diametrului de cap (aici grafic, în practică din prescrierea diametrului semifaricatului) precum și fenomenul de subțaiere la baza dinților deja amintit anterior.

În figura 30 se prezintă grafic numai profilul roții de lanț. Reprezentarea din figura 30 se obține rulând următoarea secvență:

```
close all;plot(abscl,ordl,'b','LineWidth',1.5);
axis equal;
```

Precizia (rezoluția) descrierii profilului depinde evident de

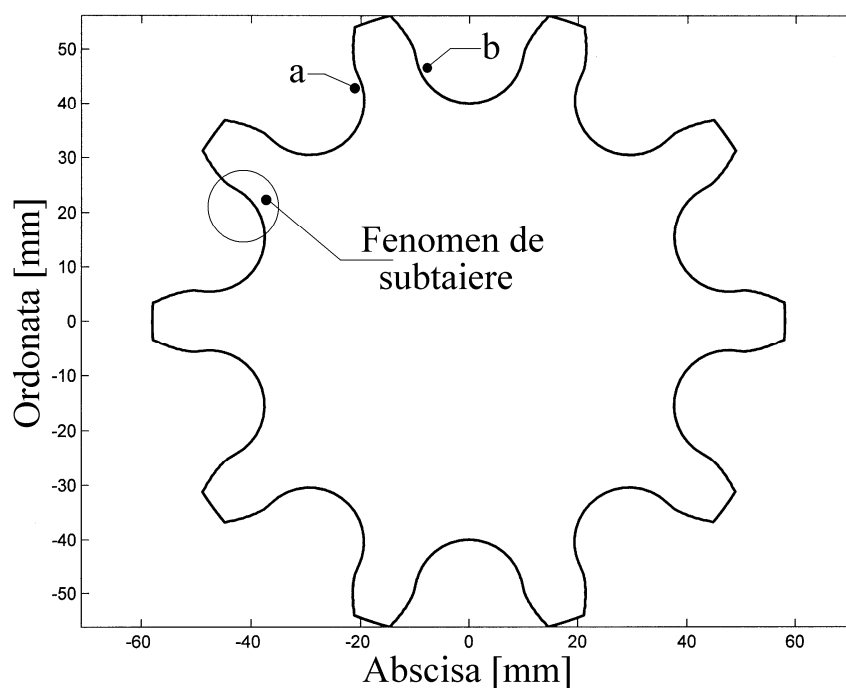


Figura 30. Reprezentare grafică a profilului roții de lanț dedus pe baza programului SIMURLAN (rezultat extras din figura 29).

rezoluția descrierii profilului generator dar și de incremenții unghiulari folosiți la descrierea pozițiilor succesive ale profilului generator respectiv la definirea sintetică a profilului. Finalitatea imediată a unei astfel de descrieri ar fi execuția roții prin frezare cu freză cilindrică (sau orice alt procedeu, de exemplu prelucrare cu fascicol LASER sau cu jet de apă sub presiune) folosind comenzi de conturare 2D.

Pe profilul din figura 30 se observă două zone distincte (**a**, **b**) cu subțaiere la baza dintelui. Admițând rotația în sens orar a profilului

generator (conform premiselor din figura 4), presupunând transpunerea acestuia pe o sculă așchietoare, subtăierea din zona **a** se produce la intrarea dintelui profilului, subtăierea din zona **b** se produce la ieșirea acestuia (așa cum rezultă și de pe figura 29). În aceeași manieră se produce subtăierea în cazul generării roților dințate cu profil evolventic.

Subtăierea, ca discontinuitate de profil a flancului de dinte în cazul roților de lanț are efecte nefaste asupra funcționării transmisiei cu lanț. Chiar în absența subtăierii să observăm necesitatea ca pe bolțurile lanțului să se plaseze role libere la rotație (cu ungere), contactul roletelor cu roata pe flacurile de dinte se face cu alunecare.

Subtăierea implică șocuri de intrare (deci zgomot) a roletelor de lanț pe roata conducătoare și de ieșire pe roata condusă. Se produce tensionare variabilă a lanțului, apariția vibrațiilor torsionale la arborii roților de lanț, deci funcționare improprie. În practica utilizării transmisiilor cu lanț s-a generalizat folosirea netensionată a ramurii inactive a lanțului (spre deosebire de transmisia cu curele, la transmisia cu lanț nu se realizează întindere). Existența acesteia atenuează aspectele negative evidențiate mai sus.

Dacă în cazul generării prin rulare a danturii evolventice subtăierea se poate elimina prin deplasare pozitivă de profil, în cazul generării roților de lanț această procedură este discutabilă. Să presupunem că o rolă oarecare a lanțului se așează într-o alveolă de pe roata de lanț, cu simetrie față de flancuri dinților adiacenți. În cazul roții generate cu deplasare de profil următoarea rolă se așează asimetric în alveolă. Asimetria se accentuează pentru următoarele role. Pentru deplasări mari de profil (necesare pentru înlăturarea subtăierii la generarea roților cu numere mici de dinți) este posibil chiar ca lanțul să nu mai fie autoghidat și să iasă de pe roată. Pentru deplasări mici de profil este probabil să apară variații ciclice ale raportului de transmitere.

Înlăturarea subtăierii (mai periculoasă pentru roțile motoare) se face cel mai comod prin creșterea numărului de dinți ai roții. Acesta este unul dintre motivele pentru care roata motoare a transmisiei cu lanț de la biciclete (de exemplu) are număr mare de dinți.

Să notăm faptul că profilul generator (conform figurii 28) folosit în generarea prin rulare a roților de lanț este foarte dificil de materializat pe sculă ca tăiș, utilizarea lui aici a avut doar suport teoretic. În practică se folosește un profil generator transformat conform figurii 31.

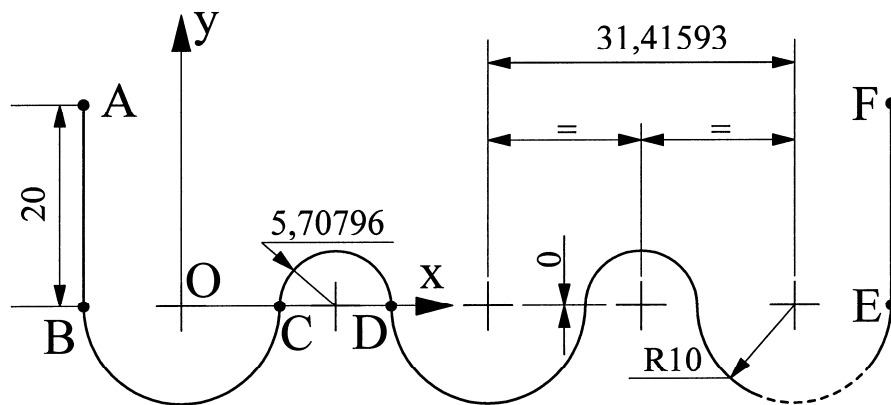


Figura 31. Elemente definitorii ale unui profil generator simplificat (asociat celui din figura 28) utilizat, utilizabil ca tăiș de sculă de mortezat,

Se folosesc numai arce semicerc BC (aferente roșilor de lanț) și arce semicerc CD ca profile de racordare. Se păstrează elementele dimensionale și de dispunere deja definite în figura 28. Profilul generator complet (extins la lungimea necesară prelucrării complete a roții), aici și în programul utilizat la simularea generării (obținut prin transformarea minoră a programului SIMURLAN), se obține prin expandarea profilului BD. Capătul profilului astfel obținut (aici notat cu E) se completează cu un punct suplimentar F, echivalentul punctului A (cu abscisa punctului E și ordonata punctului A).

Utilizarea acestui profil în simularea generării prin rulare a roții de lanț se poate face cu ajutorul programului SIMURLAN1 (redat aici și în Anexa 15), cu multe elemente de similaritate cu programul SIMURLAN.

Program SIMURLAN1

```
clear all;close all;m=10;z=17;%a=110;
```

```

fi=90*pi/180;m1=m;
rr=0.5*m*(z);dep=0*m;k=1;inc=pi/30;cons=1;
for i=pi:inc:2*pi;
x(k)=10*cos(i*cons);
y(k)=10*sin(i*cons)+dep;k=k+1;end;
for i=pi:-inc:0;
x(k)=15.7+5.70796*cos(i*cons);
y(k)=5.70796*sin(i*cons)+dep;k=k+1;end;k=k-1;
ndinti=z+5;
for j=1:ndinti;for i=1:k;
    x(j*k+i)=x(i)+j*m*pi;y(j*k+i)=y(i);
end
end
y(1)=20.7264;y(j*k+i)=20.7264;
plot(x,y);axis equal;
xc=0;yc=0;
l=1;
liminf_teta=0;limsup_teta=2.3*pi;inc=2*pi/100+0.00001;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:k*(ndinti+1)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);l=l+1;
    end
    end;l=l-1;
plot(x1,y1,'k');hold on;axis equal
axis([- (rr+1.6*m1), (rr+1.6*m1), -
(rr+1.6*m1), (rr+1.6*m1)])
xlabel('Excursia pe abscisa sistemului de
reprezentare 2D [mm]')
ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')
%rutina de determinare a profilului
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end

```



```

end
%plot(absc,ord,'r','LineWidth',1.5);
%rutina pentru limitarea diametrului
dim1=size(absc);dim=dim1(2);razlim=rr+8;
for i=1:dim
    r(i)=sqrt(abs(absc(i))^2+abs(ord(i))^2);
if r(i)>=razlim;r(i)=razlim;else end
    if ord(i)==0;ord(i)=0.000000001;else end
        if absc(i)>0;
if ord(i)>0;tetha(i)=atan(abs(ord(i))/abs(absc(i)));
else end;else end
            if absc(i)<0;if ord(i)>0;
tetha(i)=pi/2-atan(abs(ord(i))/abs(absc(i)))+pi/2;
else end;else end
                if absc(i)<0;
if ord(i)<0;
tetha(i)=atan(abs(ord(i))/abs(absc(i)))+pi;
else end;else end
                    if absc(i)>0;
if ord(i)<0;
tetha(i)=pi/2-atan(abs(ord(i))/abs(absc(i)))+3*pi/2;
else end;
else end
end
for i=1:dim;
absc1(i)=r(i)*cos(tetha(i));
ord1(i)=r(i)*sin(tetha(i));
end
plot(absc1,ord1,'k','LineWidth',1.5);axis equal

```

Singura diferență esențială față de programul SIMURLAN constă în substituirea primelor 13 linii de program (liniile L1/ până la L13/ inclusiv) cu secvența scrisă cu caractere îngroșate. Se are în vedere generarea unei roți de lanț cu 17 dinți. Secvența prezentată aici definește profilul BCD, realizează sinteza profilului generator prin expandare, și în final introduce punctele A și F.

Rularea programului SIMURLAN1 conduce la obținerea rezultatului grafic prezentat în figura 32.

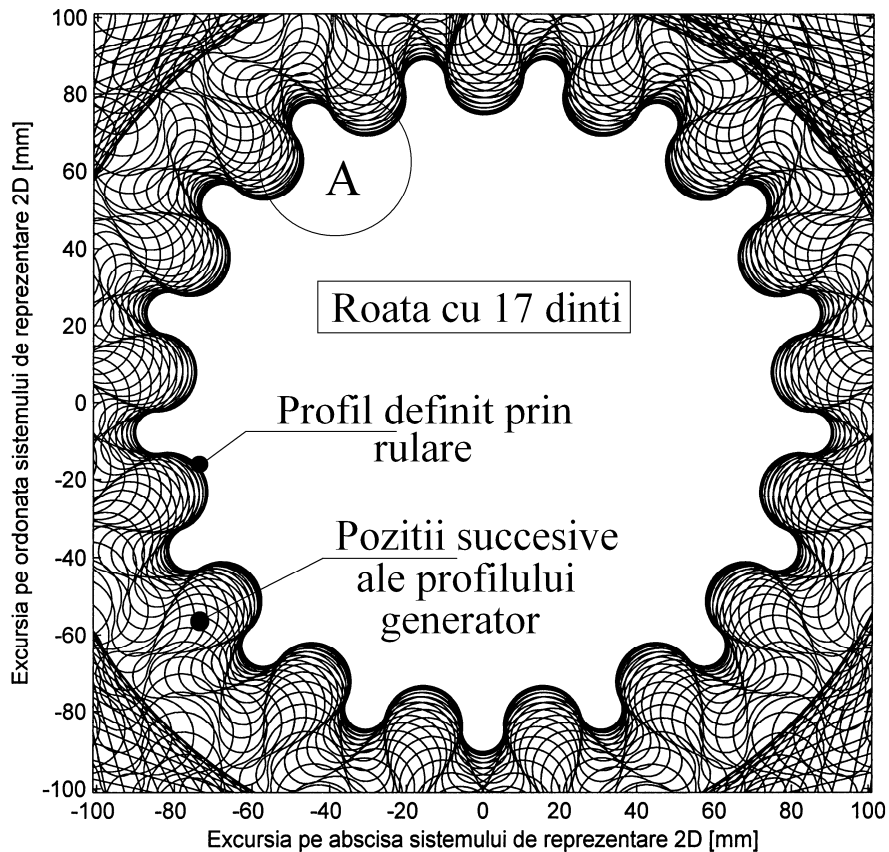


Figura 32. Rezultatul rulării programului SIMURLAN1. Definirea profilului unei roți de lanț cu 17 dinți, cu utilizarea profilului generator rezultat prin expandarea profilului definit în figura 31, fără limitarea diametrului de cap.

Pentru a reliefa mai bine aspectele rulării nu s-a procedat la limitarea fizică a diametrului de cap. Durata efectivă a simulării este semnificativ mai mare aici (cca. 1,5 ore) din cauza faptului că a intervenit suplimentar profilul curbiliniu CD, definit (ca și profilul curbiliniu BD) prin 30 de puncte distincte (crește timpul de calcul). Profilul generator apare curbiliniu pe figura 32 însă este definit ca o

succesiune de segmente (în total 1426 segmente definit prin numărul de elemente ale matricilor x sau y). Pentru reprezentarea tuturor celor 115 poziții distincte ale profilului generator s-au folosit 163.990 segmente distincte (numărul de elemente ale matricilor x_1 sau y_1).

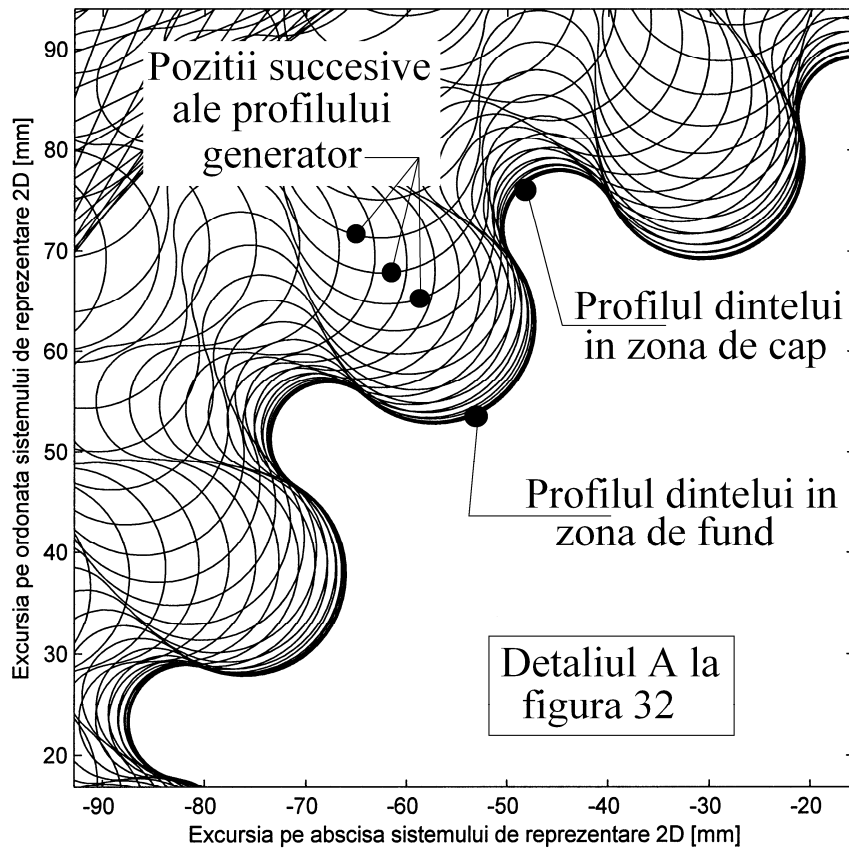


Figura 33. Detaliu grafic la figura 32 în zona A.

Pe figura 32 se observă cu claritate geometria profilului roții de lanț definit prin rulare precum și practic dispariția fenomenului de subțiere pe de o parte din cauza creșterii numărului de dinți ai roții (aici 17, pe figurile 29 și 30 roata generată are numai 10 dinți) pe de altă parte din cauza utilizării unui profil generator numai din zone

curbilinii (conform figurii 31). În figura 33 se prezintă un detaliu la figura 32 (prelevat din zona A cu ajutorul posibilităților de *zoom-in* grafic Matlab). Ca și în cazul figurii 32, se observă că profilul roții apare ca înfășurătoare a pozițiilor succesive ale profilului generator. Se

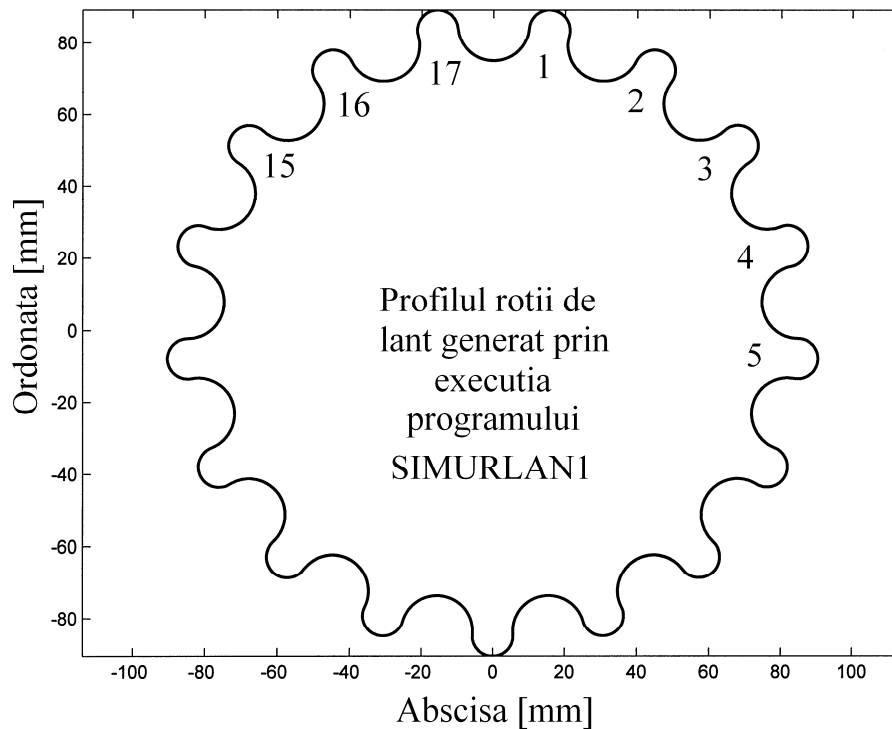


Figura 34. Extragera reprezentării profilului roții de lanț generat prin execuția programului *SIMURLAN1*.

observă de asemeni cu claritate faptul că fenomenul de subtăiere a dispărut.

Se poate extrage numai profilul generat al roții de lanț dacă după execuția programului *SIMURLAN1* se tastează în fereastra de comandă *MATLAB* instrucțiunile:

```
close all;plot(abscl,ordl,'b','LineWidth',1.5);  
axis equal;
```

Se obține reprezentarea din figura 34, de altfel redată și pe

figura 32 însă împreună cu pozițiile succesive ale profilului generator. Profilul evident cubiliniu este de fapt descris de o succesiune de 12001 segmente de dreaptă distincte (numărul de elemente ale matricilor `abscl` sau `ord1`), fiecare segment subînscrisie un unghi la centru egal cu 1,8 minute.

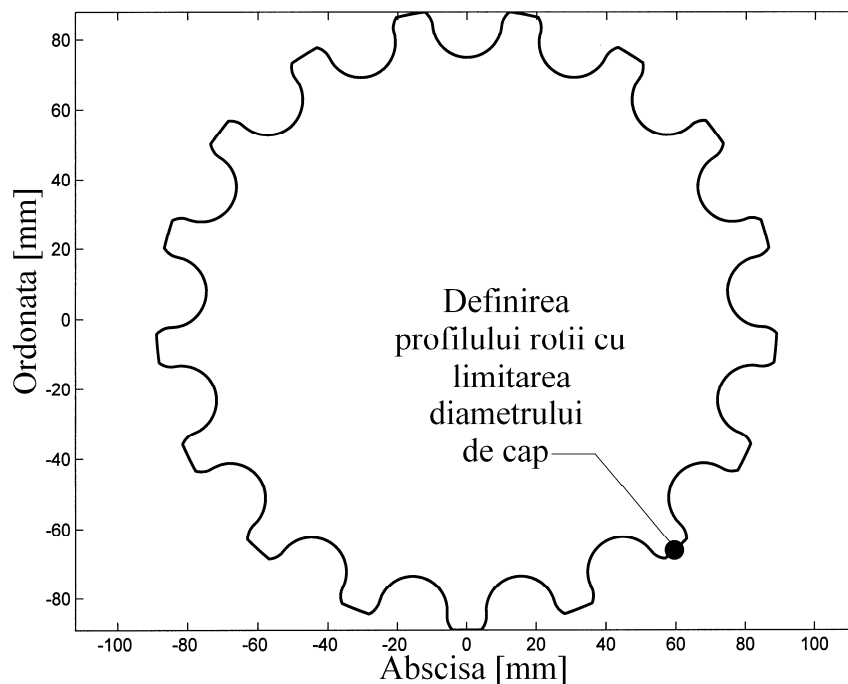


Figura 35. Extragera reprezentării profilului roții de lanț generat prin execuția programului *SIMURLAN1* cu limitarea diametrului de cap.

În practică, pentru a reduce și a simplifica implicarea în așchiere a tăișului se procedează la limitarea diametrului de cap, în maniera deja prezentată anterior. De exemplu înlocuirea instrucțiunii `razlim=rr+8;` (redată cu caractere îngroșate în listing-ul programului) cu instrucțiunea `razlim=rr+4;` și execuția programului *SIMURLAN1* permite obținerea profilului de roată de lanț redat în figura 35.

Să remarcăm faptul că limitarea diametrului de cap reduce capacitatea roții de a ghida lanțul la intrarea pe arcul de contact lanț-roată.

Să notăm o aplicație interesantă ce decurge din acest demers. Dacă profilul generator din figura 31 este înlocuit cu o structură în care arcele BC sunt materializate cu ajutorul unor role pe rulmenți, sau direct prin intermediul unor rulmenți, (să numim această structură cremalieră de rostogolire), plasarea acesteia în poziția de rulare (în angrenare) cu roata generată anterior conduce la obținerea unui echivalent al sistemului pinion-cremalieră (de convesie a mișcării de rotație în mișcare de translație sau invers). Avantajul sistemului față de cel cu cremalieră clasică constă în eliminarea frecării de alunecare.

Un demers asemănător poate fi folosit pentru generarea roților dințate cu profil în arc de cerc (neevolventic).

5. 2 Generarea prin rulare dreaptă pe cerc a roților mecanismelor cu clichet

Mecanismele cu clichet (roată-clichet) sunt folosite pentru construcția transmisiilor mecanice cu sens unic de rotație (de exemplu în construcția cheilor utilizate pentru montarea-demontarea asamblărilor filetate, sau a sistemelor cu blocare automată la rotația pe un sens, etc.). Roata de clichet (al cărei profil transversal este o curbă închisă) poate fi obținută evident prin rulare dreaptă pe cerc.

Și aici vom parcurge aceleași etape principale ale simulării, ca și anterior, după cum urmează:

1. Simularea geometrică a profilului transversal al roții de clichet, cu descrierea discretă, a coordonatelor geometrice ale punctelor esențiale ale acestui profil;
2. Obținerea profilului generator prin rulare cerc pe dreaptă. În procesul de fabricație acest profil se transferă pe o sculă așchiitoare de mortezat (sau pe o freză melc-modul).

3. Validarea simulării pe baza generării profilului transversal al roții de clichet prin rulare dreaptă pe cerc.

Fiecare dintre cele trei etape vor fi tratate pe larg după cum urmează:

5. 2. 1 Simularea profilului transversal al roții de clichet

O abordare sumară a alegerii profilului transversal ar putea conduce la concluzia că acesta poate fi conceput ca o succesiune de segmente de dreaptă. Descrierea profilului s-ar putea face foarte simplu prin coordonatele geometrice ale capetelor acestor segmente.

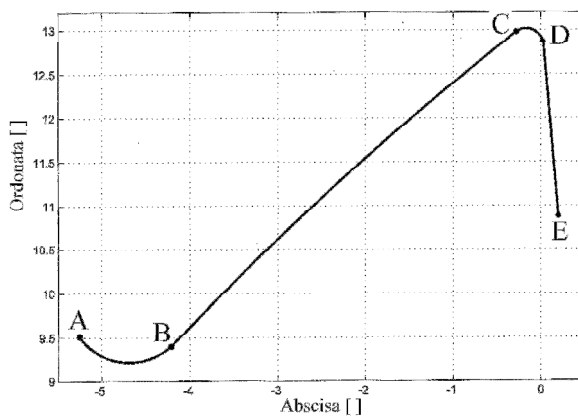


Figura 36. Descriere grafică a profilului unui dinte (cu 303 segmente, rulare a programului fig36, redat în Anexa 16).

Cu titlu de exercițiu, propunem în continuare un profil de roată de clichet cu 12 dinți, plecând de la un dinte descris prin arce de cerc și segmente de dreaptă. Se preferă folosirea arcelor de cerc pentru o mai bună repartizare a eforturilor de contact cu clichetul și pentru

reliefarea unor aspecte particulare ale rulării în această situație.

În figura 36 se prezintă profilul geometric al unui dinte al roții de clichet, obținut prin rularea programului Matlab **fig36**. Profilul de dinte a fost proiectat în AutoCAD după care a fost transpus în mediul Matlab pentru calculul coordonatelor punctelor esențiale.

Program fig36

```
close all;clear all;tic;
%GENERARE COORDONATE PUNCTE CARACTERISTICE DE
DEFINIRE A ARCULUI DE CERC DINTRE PUNCTELE A ȘI B
nincr=100;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
```

```

limunghi1=215*pi/180;limunghi2=limunghi1+(101.3334*pi
i/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end

%GENERARE COORDONATE PUNCTE CARACTERISTICE DE
DEFINIRE A ARCULUI DE CERC DINTRE PUNCTELE B ȘI C
for i=0:nincr;constx1=22.71035;consty1=-16.23102;
    limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);
    k=k+1;
end

%GENERARE COORDONATE PUNCTE CARACTERISTICE DE
DEFINIRE A ARCULUI DE CERC DINTRE PUNCTELE C ȘI D
for i=0:nincr;constx1=-0.16718;consty1=12.81608;
    limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
    k=k+1;
end

%GENERAREA COORDONATELOR PUNCTULUI E
x(k)=0.20356;y(k)=10.87329;num=k;

%DESENARE A PROFILULUI DE DINTE
plot(x,y,'k','LineWidth',2);axis equal;grid
axis([-5.5 0.5 9 13.2])

```

Pe listing-ul de program sunt delimitate zonele aferente generării fiecăruia dintre arcele de cerc AB, BC și CD. Fiecare arc de cerc este de fapt aproximat prin câte o succesiune de 100 de segmente de dreaptă (variabila $nincr=100$). Segmentul de dreaptă DE este definit prin coordonatele punctelor D și E. Arcul AB descrie profilul de

racordare dintre flancurile dintelui în zona de picior. Arcul CD descrie profilul de racordare dintre flancurile dintelui în zona de cap. Toate coordonatele sunt descrise în raport cu centrul roții în formă carteziană (abscisă, ordonată).

Pentru generarea și reprezentarea grafică a coordonatelor carteziene ale tuturor punctelor de definire a profilului transversal al roții de clichet se rotește dintele reprezentat în figura 36 de 11 ori în jurul originii.

Mai întâi se determină coordonatele în formă polară (rază, unghi) ale punctelor implicate în definirea unui dinte, prin rularea următoarei secțiuni de program:

```
for i=1:num;
    unghi(i)=pi+atan(y(i)/x(i));
    if x(i)>0;unghi(i)=atan(y(i)/x(i));else end
    raza(i)=sqrt(x(i)^2+y(i)^2);
end
```

Apoi aceste coordonate polare sunt folosite pentru calculul tuturor punctelor profilului transversal al roții (rotire de 11 ori în jurul originii) prin rularea secțiunii de program:

```
clear x;clear y;k=1;
for j=0:11;
    for i=1:num;
        x(k)=raza(i)*cos(unghi(i)-pi/6*j);
        y(k)=raza(i)*sin(unghi(i)-pi/6*j);
        k=k+1;
    end
end
plot(x,y,'k','LineWidth',1.5);axis equal;grid
axis([-13.5 13.5 -13.5 13.5])
```

care are drept finalitate imediată reprezentarea completă a profilului transversal de roată de clichet, redat grafic în figura 37. Profilul este redat ca succesiune de 3647 segmente. Aproximarea arcelor de curbă prin segmente de dreaptă este evident un compromis tehnic care facilitează simularea, în primul rând determinarea numerică a coordonatelor profilului generator regăsit pe sculă dar și a coordonatelor profilului transversal al roții de lanț generat prin rulare, aspecte pe deplin valorificate anterior.

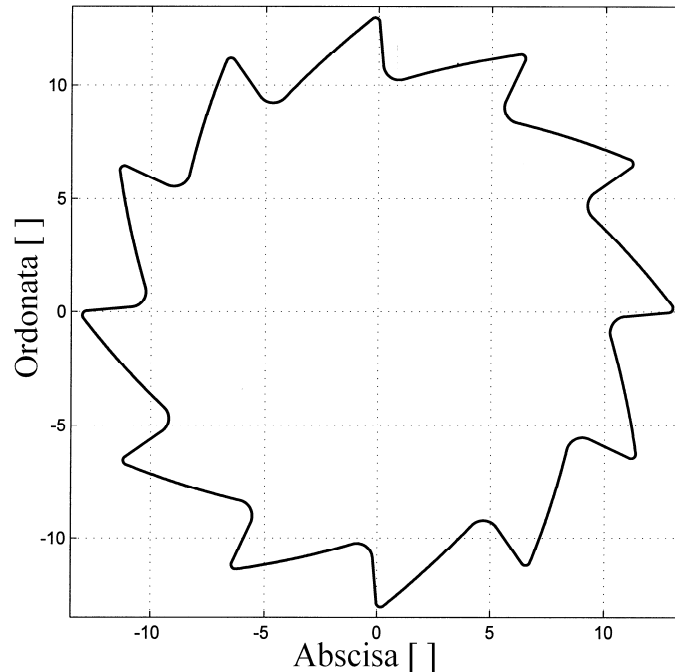


Figura 37. Descriere grafică a profilului transversal teoretic al roții de clichet (cu 3647 segmente, rulare a programului fig37, Anexa 17).

Acest profil asigură blocarea roții de clichet pentru sensul de rotație orară (*Clockwise*). Profilul simetric față de axa ordonatelor (care asigură blocarea pentru sensul de rotație antiorară, *Counter Clockwise*) se obține dacă instrucțiunea de desenare din secvența anterioară este precedată de instrucțiunea $x=-x$;

5. 2. 2 Obținerea profilului generator al roții de clichet

Profilul generator este obținut prin rularea cerc pe dreaptă a profilului roții de lanț descris în figura 37. Raza cercului de rulare se consideră a fi raza maximă a profilului roții (13,01717 mm). Se aplică elementele teoretice deja prezentate anterior, concretizate în sinteza programului DETPROFB, cu listing prezentat în continuare (reluat în Anexa 18). Sunt inserate o serie de comentarii cu majuscule care facilitează înțelegerea secțiunilor de program:

Program DETPROFB

```

close all;clear all;tic;
%AICI INCEPE SECTIUNEA DE DEFINIRE A PROFILULUI
TEORETIC A ROTII DE CLICHET NECESAR DEFINIRII
PROFILULUI GENERATOR PRIN RULARE CERC PE DREAPTA
r=13.01717;a=100;b=0;nincr=30;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
limunghi1=215*pi/180;limunghi2=limunghi1+(101.3334*pi
i/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=22.71035;consty1=-16.23102;
limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);k=k+1;
end
for i=0:nincr;constx1=-0.16718;consty1=12.81608;
limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
k=k+1;
end
x(k)=0.20356;y(k)=10.87329;num=k;
%x1=x;y1=y;
    for i=1:num;
        unghi(i)=pi+atan(y(i)/x(i));
        if x(i)>0;unghi(i)=atan(y(i)/x(i));else end
        raza(i)=sqrt(x(i)^2+y(i)^2);
    end
clear x;clear y;k=1;
for j=0:11;

```

```

for i=1:num;
    x(k)=raza(i)*cos(unghi(i)-pi/6*j);
    y(k)=raza(i)*sin(unghi(i)-pi/6*j);k=k+1;
end
end
%plot(x,y);axis equal;xdes=x;ydes=y;
k=k-1;l=1;
x=-x;
%AICI INCEPE SECTIUNEA IN CARE PROFILUL TEORETIC AL
ROTII ESTE RULAT (CERC PE DREAPTA) DEFININD PROFILUL
GENERATOR CA INFASURATOARE A POZITIILOR SUCCESIVE
ALE ROTII
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end;l=l-1;
plot(x1,y1,'k');hold on;axis equal;hold on
%AICI INCEPE SECTIUNEA DE DETERMINARE A
COORDONATELOR PUNCTELOR INFASURATOAREI POZITIILOR
SUCCESIVE ALE ROTII
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=-20:inc2:100;miny=1000;ics=0;abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;
                else end;else end;else end

```

```

        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;miny=int;
else end;else end;else end;end;l=l+1;
if miny<minminy;minminy=miny;
else end;end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'k','LineWidth',1.5);axis equal
%AICI INCEPE SECTIUNEA DE DETERMINARE A
COORDONATELOR PROFILULUI GENERATOR
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
%axis([0,30,-15,15]);toc

```

Rezultatul grafic al rulării este reprodus în figura 38. Există suficient de multe similitudini cu rezultatele deja obținute anterior (figurile 3 și 17).

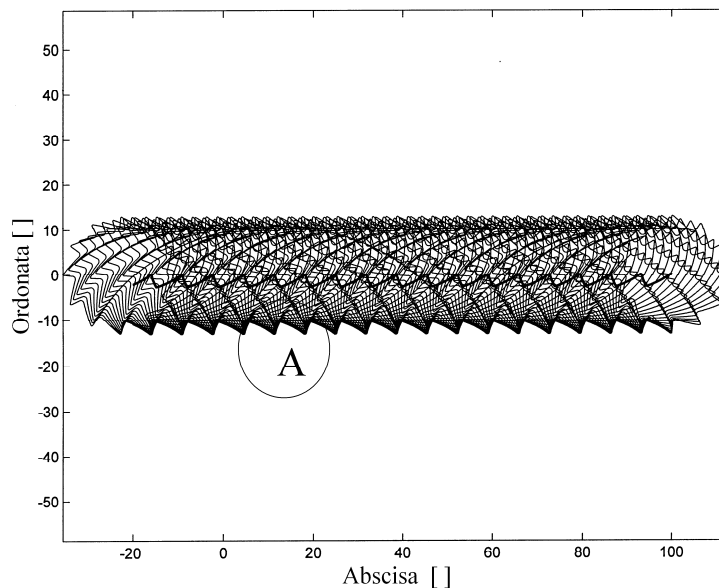


Figura 38. Rezultat grafic al rulării programului *DETPROFB*.
Determinarea profilului generator (Anexa 18).

Dacă se activează ultima instrucțiune a programului *DETPROFB* (`axis([0,30,-15,15]);toc`) atunci se obține un detaliu grafic al figurii 38 menit să evidențieze mai clar elementele implicate în definirea profilului generator. Acest detaliu este redat în

figura 39. Pentru scurtarea duratei de execuție a programului s-a redus la 30 numărul de segmente de dreaptă folosite în descrierea fiecăruia dintre arcele de cerc AB, BC și DE (variabila $n_{incr}=30$) din

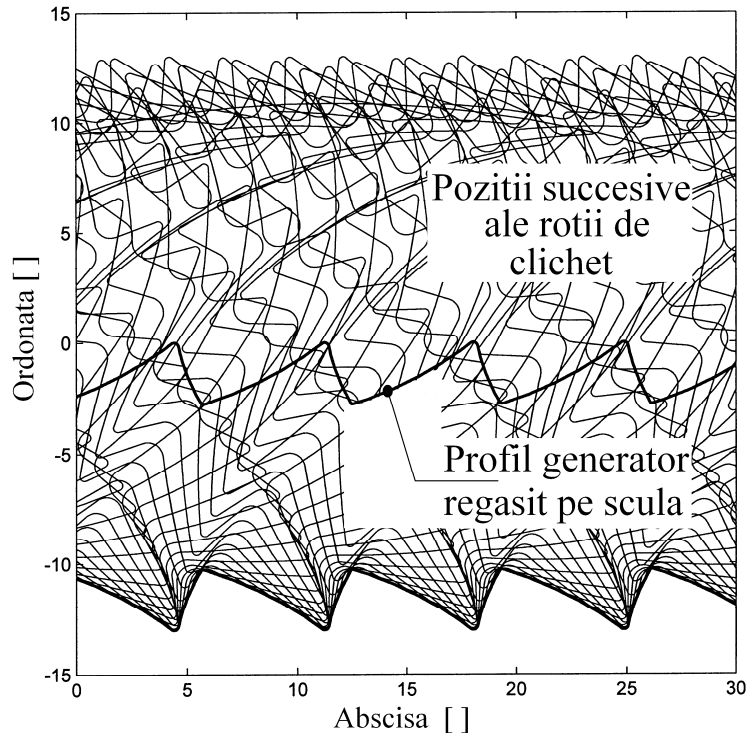


Figura 39. Detaliu grafic la figura 38 din zona A.

descrierea profilului de dinte (din figura 36). În aceste condiții execuția programului se face în 199 secunde. Altfel (cu variabila $n_{incr}=100$) durata de execuție este de 39 minute, cu mult mai mare decât anterior dar cu avantajul unei definiri mai bune a profilului generator. O definire și mai bună a acestui profil se poate obține dacă incrementul unghiular de rostogolire la rularea cerc pe dreaptă din instrucțiunea `for i=0:3*pi/100:3*pi;u=-i;` (aici de $3,6^\circ$ de arc) este redus și mai mult. Evident, aceasta duce la creșterea suplimentară a duratei de execuție a programului.

Dacă acum se introduce în fereastra de comandă Matlab setul de instrucțiuni următor:

```
close all;
```

```
plot(abs,ord,'k','LineWidth',1.5);
axis([2,14,-3,.6])
```

atunci la execuția acestora se obține reprezentarea grafică detaliată a unei porțiuni din profilul generator, conform figurii 40.

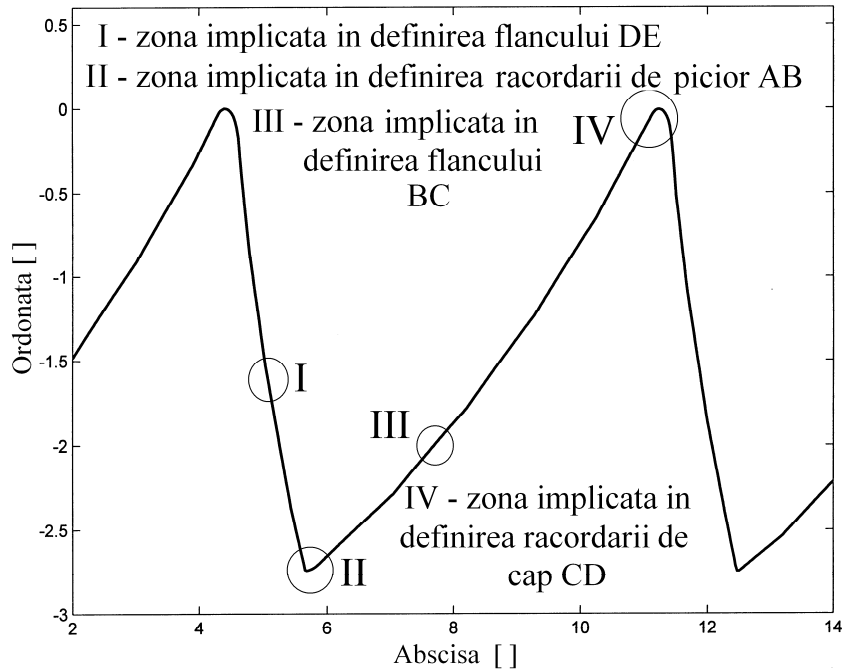


Figura 40. Reprezentare grafică a unei porțiuni din profilul generator.

Pe figură s-au evidențiat zonele implicate în definirea fiecărei porțiuni a profilului de dinte pe piesă.

5. 2. 3 Validarea simulării. Generarea profilului transversal al roții de clichet prin rulare dreaptă pe cerc

Profilul generator obținut anterior, descris ca succesiune de segmente de dreaptă, fiecare segment fiind delimitat de puncte cu coordonatele rezidente în matricile *abs* (pentru abscise) și *ord* (pentru ordonate), poate fi utilizat în continuare pentru validarea simulării. Simularea rulării dreaptă pe cerc (cu profilul generator solidar cu

dreapta) trebuie să genereze într-un plan solidar cu cercul exact profilul transversal al roții de clichet (descries grafic în figura 37) avut în vedere la începutul demersului nostru. Dacă după rularea programului DETPROFB se introduce în fereastra de comandă și se execută secvența de program sistematic utilizată anterior și reluată mai jos:

```
close all;clear x1;clear y1;clear x;clear y;
x=abs+15.86;y=ord;rr=r;fi=pi/2;xc=0;yc=0;
liminf teta=0;limsupteta=3*pi;inc=2*pi/300;
for teta=liminf teta:inc:limsupteta;
    for i=1:dim
        x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-teta-
        pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-teta-
        pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-1.2*r),(yc+1.2*r)])
```

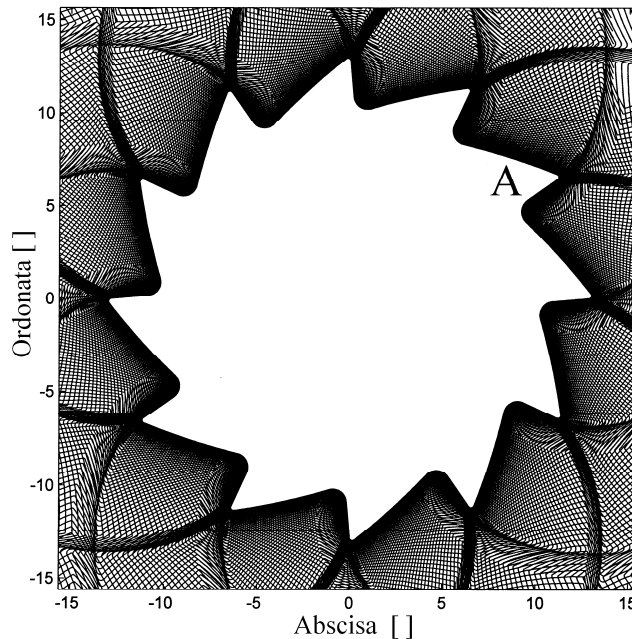


Figura 41. Descriere grafică a rezultatului rulării pentru definirea profilului de roată de lanț (rezultat rulare programul CLICHET1, Anexa 19).

atunci se obțin și se redau grafic pozițiile succesive ale profilului generator generate prin rulare dreapta pe cerc, conform figurii 41. Peste reprezentarea aceasta s-a suprapus și profilul teoretic al roții de clichet, cel de la care s-a plecat inițial (descriș în figura 37), redat cu linie groasă, desenat cu instrucțiunea:

```
plot(xdes,ydes,'k','LineWidth',1.5)
```

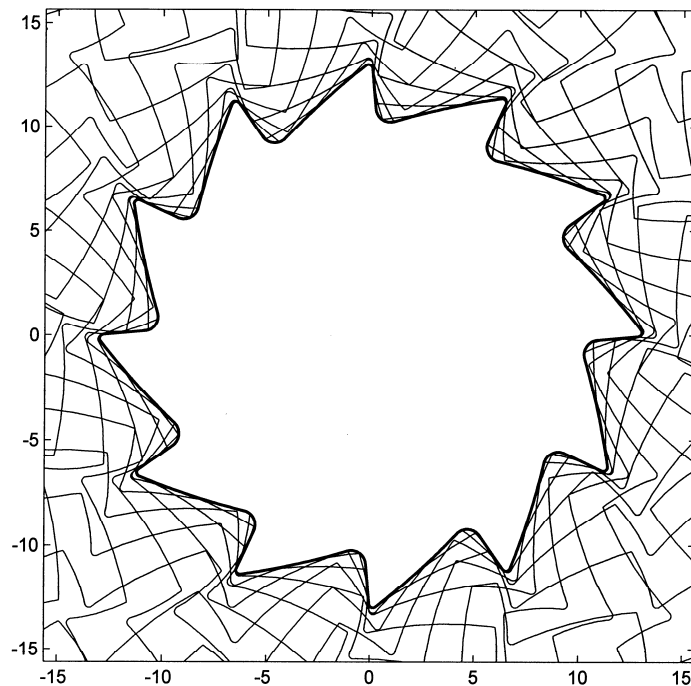


Figura 42. Redare a figurii 41. Rulare cu increment unghiular majorat
($inc=2*\pi/30;$)

Pentru a asigura o bună suprapunere cu profilul teoretic, s-a rotit convenabil figura generată prin rulare cu ajutorul instrucțiunii $x=abs+15.86;$. Rotirea figurii s-a realizat prin deplasarea către dreapta a profilului generator, înainte de începerea procesului de rulare. Secțiunile de program care au concurat la obținerea figurii 41 sunt reunite în programul CLICHET1 redat în Anexa 19.

Cel puțin la o primă evaluare pe figura 41, profilul teoretic apare ca perfect înfășurat de pozițiile succesive ale profilului generator.

Aspectul se poate constata mai bine în figura 42 care descrie grafic rezultatul rulării programului CLICHET1 folosind un increment unghiular mai mare pentru unghiul θ (în program variabila *teta*), cu $inc=2*\pi/30$). Spre comparație, pentru obținerea figurii 41 s-a utilizat un increment de zece ori mai mic ($inc=2*\pi/300$). Un detaliu al figurii 41 din zona A a unui dinte al roții (redat în figura 43) se poate obține dacă, după rularea normală a programului CLICHET1, se execută instrucțiunea: `axis([5 12 3 11])`.

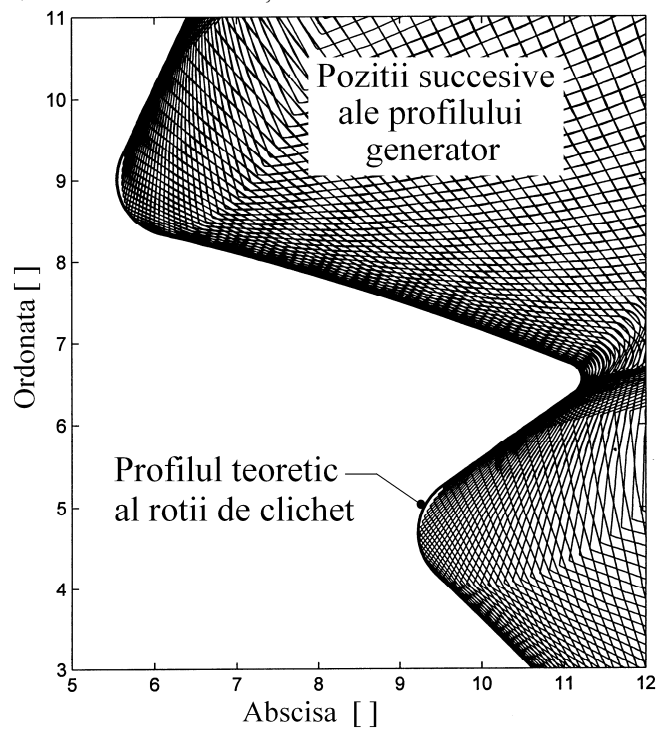


Figura 43. Detaliu grafic la figura 41 (în zona A).

Se observă faptul că profilul teoretic diferă cu foarte puțin de cel definit prin rulare doar în zona de picior a dintelui (sub 0,1 mm diferență).

Acesta este un rezultat foarte important pentru practică în raport cu considerațiile de la § 3.2.1, zona de picior este foarte puțin afectată datorită viciului de rulare semnalat anterior (prin arc de trecere între flancuri, figurile 16 și 21) dacă pe profilul teoretic flancurile vecine ale dintelui sunt racordate prin arc de cerc în zona de picior. Se poate efectua un studiu care să deducă natura geometrică a curbei de

racordare în zona de picior (alta decât arcul de cerc) care să elimine complet viciul de rulare.

O confirmare clară a acestei concluzii este oferită de rezultatul reliefat în figura 44. S-a refăcut tot demersul de până acum, referitor la o roată de clichet formal identică, singura diferență fiind profilul teoretic al dinților, de această dată flancurile sunt rectilinii, fără racordare la picior și la cap.

Pe considerentele care au condus la sinteza programului CLICHET1 (redat în Anexa 19) s-a conceput programul CLICHET2 (redat în Anexa 20) pentru roată cu profil teoretic cu flancuri rectilinii. Diferențele dintre aceste două programe sunt minore. Pot fi parcurse aceleași etape care au condus la obținerea figurilor 36 ÷ 43. Se prezintă aici numai detaliul grafic echivalent, obținut în condițiile figurii 43, conform figurii 44.

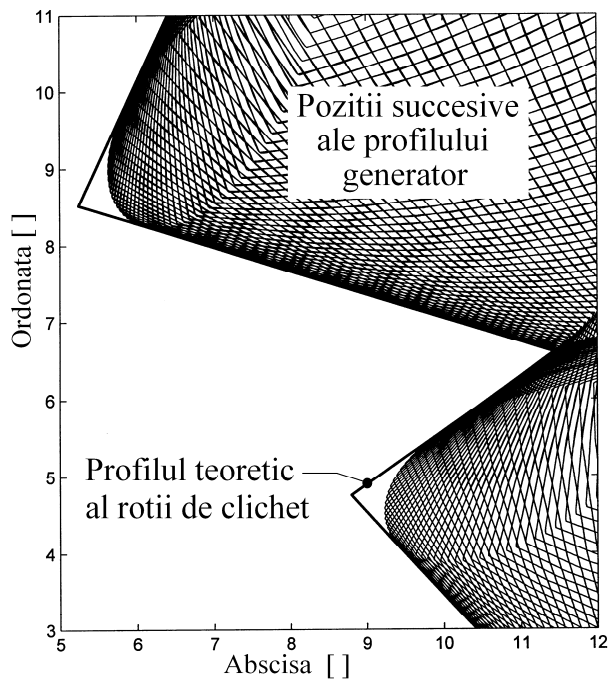


Figura 44. Echivalent grafic al figurii 43 în cazul generării prin rulare a unei roți de clichet cu flancuri rectilinii, fără racordări.

Este evident faptul că, în acest caz, forma vârfului de dinte nu este afectată (aspect deja ilustrat anterior în figurile 16 și 21) dar zona de picior este puternic viciată.

Se pot acum întreprinde demersuri teoretice pentru sinteza unui program pentru determinarea coordonatelor geometrice ale profilului înfășurat prin rulare (regăsit ca secțiune transversală a roții prelucrate) aplicând de exemplu considerente identice cu cele folosite în cazul figurii 35 (aplicate în programul SIMURLAN1 din Anexa 15).

6. Aspecte ale simulării rulării cerc pe cerc

Un neajuns esențial al materializării în fabricație (prin mortezare) a rulării dreaptă pe cerc este faptul că scula pe care se regăsește profilul generator are lungime finită, în timp ce rularea pentru prelucrarea completă a unei piese presupune utilizarea unei lungimi de sculă foarte mare. În practică acest neajuns se rezolvă prin așa numita divizare discontinuă: procesul de rulare se oprește iar scula se retrage radial, se translează cu un număr întreg de pași în sens opus celui utilizat la rulare, este repositionată după care procesul se reia.

Este și acesta un neajuns care se poate rezolva foarte ușor dacă se utilizează un proces de fabricație prin rulare de manieră ușor diferită: așa numita rulare cerc pe cerc (și rulanta și baza sunt cercuri în contact). Avantajul esențial este acela că profilul generator rezultă ca o curbă închisă, regăsită ca tăiș pe sculă. Scula este practic un cuțit roată de mortezat, cu mișcare de rotație, care își transferă prin rulare profilul generator (solidar cu rulanta) pe piesă. Profilul astfel generat pe piesă este solidar cu baza. Profilul generat împreună cu curba directoare (de regulă segment de dreaptă, generat prin mișcare de mortezare, rectilie alternativă) definesc suprafața prelucrată pe piesă.

Rulanta și baza (ambele cercuri) rulează. Baza este fixă iar rulanta se rostogolește peste aceasta (se rotește în jurul propriului centru dar și în jurul centrului bazei), are contact punctual cu baza. Punctul de contact este centru instantaneu de rotație pentru rulanta. Evident, rularea se poate produce cu ambele cercuri în mișcare de rotație în jurul propriilor centre cu condiția păstrării unui contact punctual (punct de tangență) în care viteza tangențială a celor două cercuri este aceeași. Acest tip de rulare se folosește în fabricație, procedeul de mortezare cu cuțit roată de mortezat.

Pentru abordările din acest capitol vom considera rularea cu baza fixă.

Obținerea unui profil oarecare de tip curbă închisă prin rulare presupune parcurgerea etapelor deja enumerate anterior:

- definirea teoretică a profilului de generat;
- definirea profilului generator regăsit pe scula de mortezat;
- obținerea profilului de generat prin rularea profilului generator (transferul de pe sculă pe piesă);

Fără a restrânge generalitatea abordării, vom considera că profilul de generat este deja definit prin curbe simple (aici cu preponderență contururi de tip poligon regulat). Cititorul interesat va putea exersa acest demers pentru orice alt tip de profil curbiliniu închis.

6. 1. Definirea profilului generator

Se poate face foarte simplu prin rularea cerc pe cerc dacă se consideră că profilul de generat este solidar cu rulanta. Profilul generator va fi definit ca înfășurătoare a pozițiilor succesive ale profilului de generat într-un plan solidar cu baza.

Vom considera că profilul de generat este un triunghi echilateral înscris într-un cerc cu raza de 50 de unități.

Vom considera inițial că cercul rulantă și cercul bază au raze egale între ele și cu cea de mai sus.

Simularea rulării se face cu ajutorul programului TRIUNGHI2, cu listing-ul prezentat în continuare (și reluat în Anexa 21). Se rostogolește cercul rulantă peste cercul bază (care aici este fix). Dacă razele celor două cercuri sunt egale, atunci rostogolirea presupune că dacă centrul rulantei se rotește într-un sens oarecare cu un unghi α în jurul centrului bazei, atunci rulanta se rotește cu un unghi $\beta = 2 \cdot \alpha$ în același sens. În simulare se folosește sensul de rotație trigonometric.

Program TRIUNGHI2

```
clear all;Close all;tic
nlob=3;%nlob este numarul de lobi al sculei
%cu nlob=3 cercul baza si cercul rulanta au aceeasi
raza
rrul=50;rbaz=rrul*nlob/3;xc=0;yc=0;
x(1)=-43.3013;y(1)=-25;
x(2)=43.3013;y(2)=-25;
```



```
x(3)=0;y(3)=50;
x(4)=-43.3013;y(4)=-25;
k=4;
```

**%AICI INCEPE SINTEZA PROFILULUI GENERATOR REGASIT PE
SCULA PRIN RULARE CERC PE CERC**

```
incr=2*pi/(500*nlob);
alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+3)/3*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);

y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*si
n(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
x1a=x1;y1a=y1;
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
```

**%SECTIUNE DE DETERMINARE A COORDONATELOR GEOMETRICE
ALE PUNCTELOR PROFILULUI GENERATOR**

```
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
```

```

if dif1==0;dif1=0.00000000001;else end;
    intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
    dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
    if dist1<dist;
        if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else end;
        if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
        if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
plot(absc,ord,'k','LineWidth',1.5);axis equal;hold
on
plot(x1a,y1a,'k','LineWidth',1.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
xcerc(l)=(rbaz+rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)
=(rbaz+rrul)*sin(alfa1)+rrul*sin(i);

```

```

l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE PE
PROFILUL GENERATOR, REGASIT PE TAISUL SCULEI DE
MORTEZAT

```

Rezultatul rulării acestui program se prezintă în figura 45.

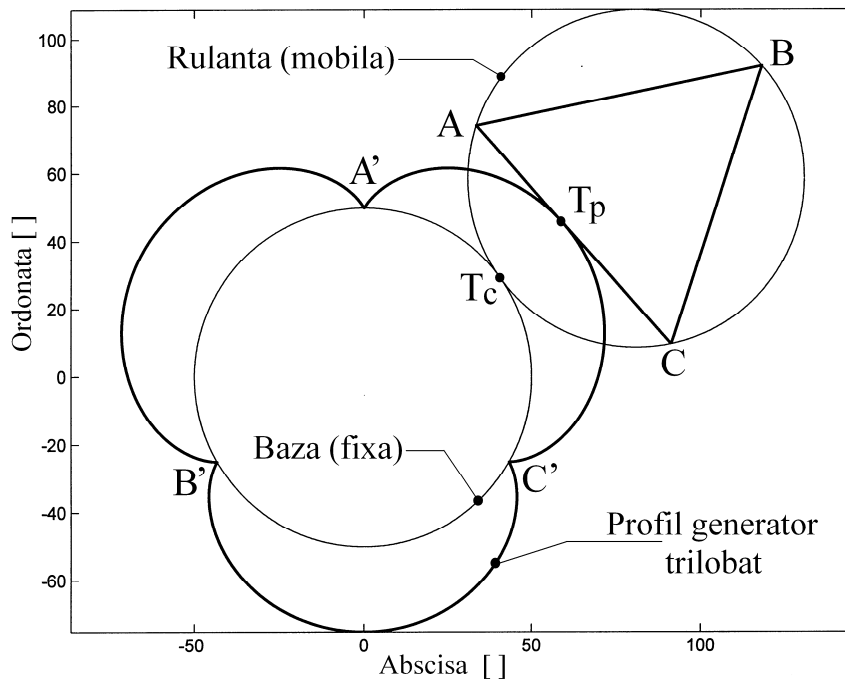


Figura 45. Rezultat grafic al execuției programului TRIUNGHI2. Redare grafică a schemei de generare a profilului generator cu trei lobi.

Sunt reprezentate aici:

- descrierea grafică a profilului generator trilobat obținut prin rularea triunghiului;
- cercul bază;
- o poziție a cercului rulantă împreună cu triunghiul (cu titlu de exemplu).

Profilul generator este definit ca o succesiune de 12001 segmente tangente la profilul teoretic, coordonatele capetelor acestora (care pot fi considerate practic ca puncte ale profilului) se regăsesc în matricile absc (abscisele) și ord (ordonatele). Pe figura 45 se observă

faptul că puncte aflate pe rulantă (vârfurile triunghiului A, B, C) generează puncte aflate pe bază (A', B', C'). Este evidentă coliniaritatea centrelor bazei și a rulantei cu punctul de tangență a acestora (T_c) și punctul de tangență a profilelor (T_p). La fel de evident este faptul că, spre deosebire de punctul T_c , în punctul T_p există alunecare între triunghi și profilul generator.

Dacă după execuția programului TRIUNGHI2 se introduce în fereastra de comandă și se execută următorul set de instrucțiuni:

```
close all;
plot(x1,y1,'k','LineWidth',1);
hold on;
axis equal
plot(x1a,y1a,'w','LineWidth',2);
```

Se obține reprezentarea din figura 46, a tuturor pozițiilor (6413) triunghiului (și o poziție generică a acestuia). Aceste poziții definesc prin înfășurare (la interior) profilul generator descris grafic în figura 45.

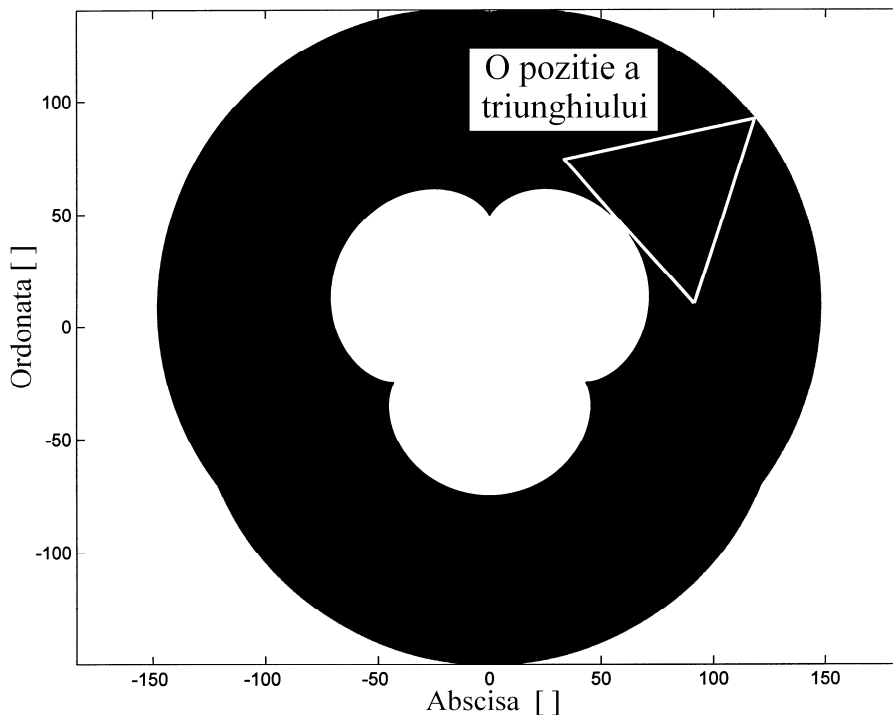


Figura 46. Reprezentare a pozițiilor succesive ale triunghiului. Definirea profilului generator prin înfășurare.

Este interesant de remarcat faptul că nu este obligatoriu ca raza cercului bază (R_b) să fie egală cu cea a cercului rulantă (R_r). Cele două raze pot fi diferite dacă se realizează condiția matematică:

$$(14) \quad N_A \cdot R_b = N_B \cdot R_r$$

În (14) N_A și N_B sunt numere întregi. Pentru profile de generat sub forma unui poligon (obligatoriu regulat), N_A reprezintă numărul de laturi (aici, pentru un triunghi, $N_A = 3$). Pentru profile regulate de altă formă (roată dințată, arbore canelat, etc.) N_A este egal cu numărul de dinți (caneluri).

Profilul generator va avea o formă care depinde de profilul de generat dar și de N_B . Pentru profile de generat neregulate, obligatoriu $N_A = 1$.

Caracteristicile profilului generator pentru diferite valori ale constantei N_B se poate obține rulând programul TRIUNGHI2. Valoarea constantei N_B se prescrie în linia 2 (variabila `nlob`).

Condiția rulării cu cercul bază fix se poate enunța după cum urmează:

- dacă centrul rulantei se rotește față de centrul bazei cu unghiul α în sens orar atunci rulanta se rotește în jurul propriului centru cu unghiul $\beta = \alpha \cdot (N_A + N_B) / N_A$ în sens orar.

Dacă rularea se produce cu centrul cercului rulantă fix (cel mai adesea în fabricația prin mortezare) atunci condiția rulării poate fi reformulată după cum urmează:

- dacă cercul bază se rotește în jurul propriei axe cu unghiul α în sens orar, atunci cercul rulantă se rotește în jurul propriei axe, în sens antiorar cu unghiul $\beta = N_B / N_A \cdot \alpha$. Raportul turațiilor celor două cercuri este riguros determinat de relația:

$$(15) \quad n_r / n_b = \beta / \alpha = N_B / N_A$$

Figura 45 este generată pentru valoarea $N_B=3$ (și evident, $N_A=3$).

În figura 47 se prezintă schema de generare și profilul generator cu un singur lob ($N_B=1$, $N_A=3$). Pentru aceasta se rulează programul TRIUNGHI2 în care se prescrie valoarea variabilei `nlob=1`.

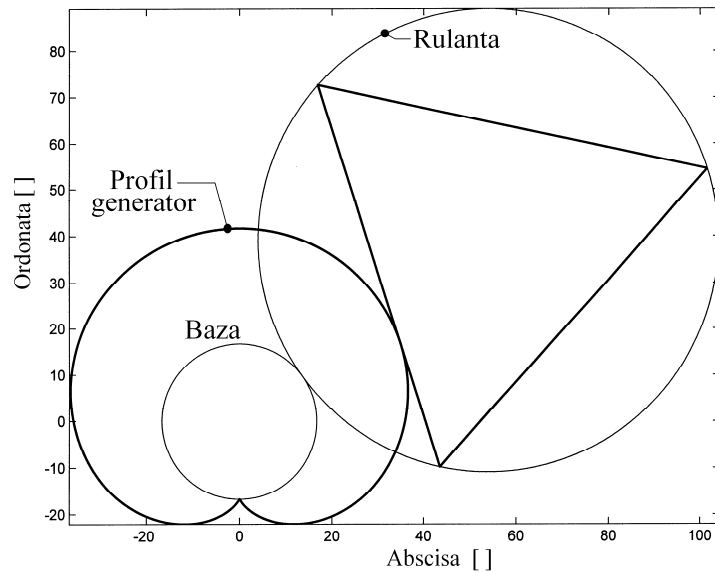


Figura 47. Rezultat grafic al execuției programului TRIUNGHI2 cu $N_B=1$ (variabila $nlob=1$). Redare grafică a schemei de generare a profilului generator cu un singur lob.

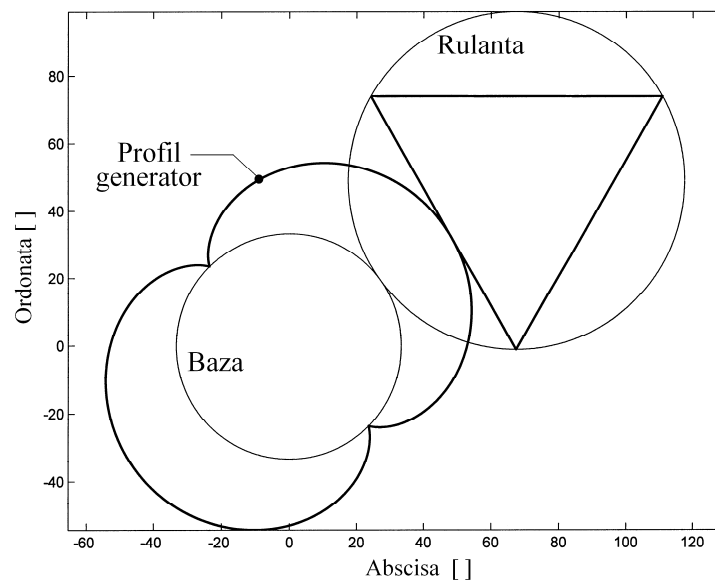


Figura 48. Rezultat grafic al execuției programului TRIUNGHI2 cu $N_B=2$ (variabila $nlob=2$). Redare grafică a schemei de generare a profilului generator cu doi lobi.

În figura 48 se prezintă schema de generare și profilul generator cu doi lobi ($N_B=2$, $N_A=3$). Pentru aceasta se rulează programul TRIUNGHI2 în care se prescrie valoarea variabilei $nlob=2$.

În figura 49 se prezintă schema de generare și profilul generator cu patru lobi ($N_B=4$, $N_A=3$). Pentru aceasta se rulează programul TRIUNGHI2 în care se prescrie valoarea variabilei $nlob=4$.

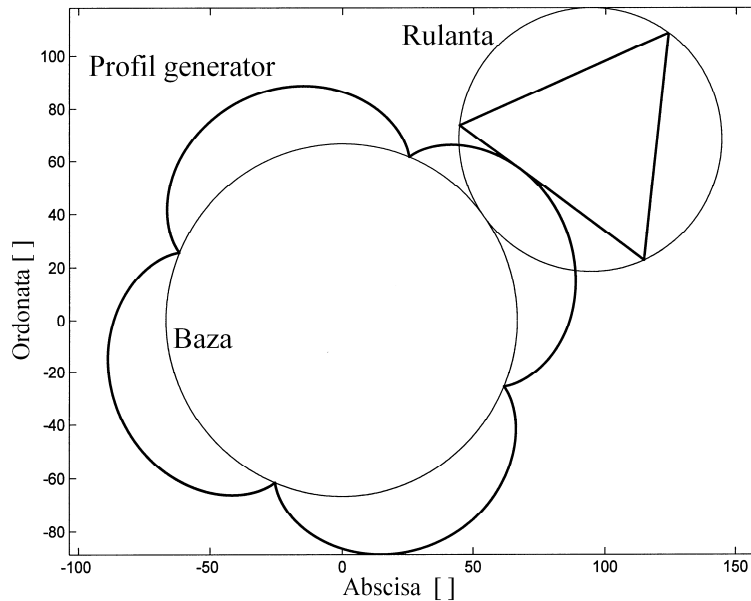


Figura 49. Rezultat grafic al execuției programului TRIUNGHI2 cu $N_B=4$ (variabila $nlob=4$). Redare grafică a schemei de generare a profilului generator cu patru lobi.

Evident, exercițiul se poate continua pentru orice număr de lobi ai profilului generator. În practică se recomandă utilizarea unei scule cu număr redus de lobi.

Programul TRIUNGHI2 poate fi adaptat pentru obținerea profilului generator destinat generării prin rulare a oricărei curbe plane închise. De exemplu în cazul unui pătrat, dacă în prima parte a programului (până la instrucțiunea $k=4$ inclusiv) se înlocuiește cu setul de instrucțiuni:

```
clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN RULARE
CERC PE CERC
```

```

nlob=3;%nlob este numarul de lobi al sculei
%cu nlob=4 cercul baza si cercul rulanta au aceeasi
raza
r=10.6066*2^0.5;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
k=5;

```

iar instrucțiunea $\text{beta1}=\text{beta}+(\text{nlob}+3)/3*i$; se înlocuiește cu :
 $\text{beta1}=\text{beta}+(\text{nlob}+4)/4*i$; Atunci se obține un program ce
 poate fi numit PATRAT2 (redat în Anexa 21 a). Rularea acestuia
 permite determinarea profilului generator pentru $N_B=3$ (de exemplu) și
 evident $N_A=4$, cu generarea rezultatelor conform figurii 50.

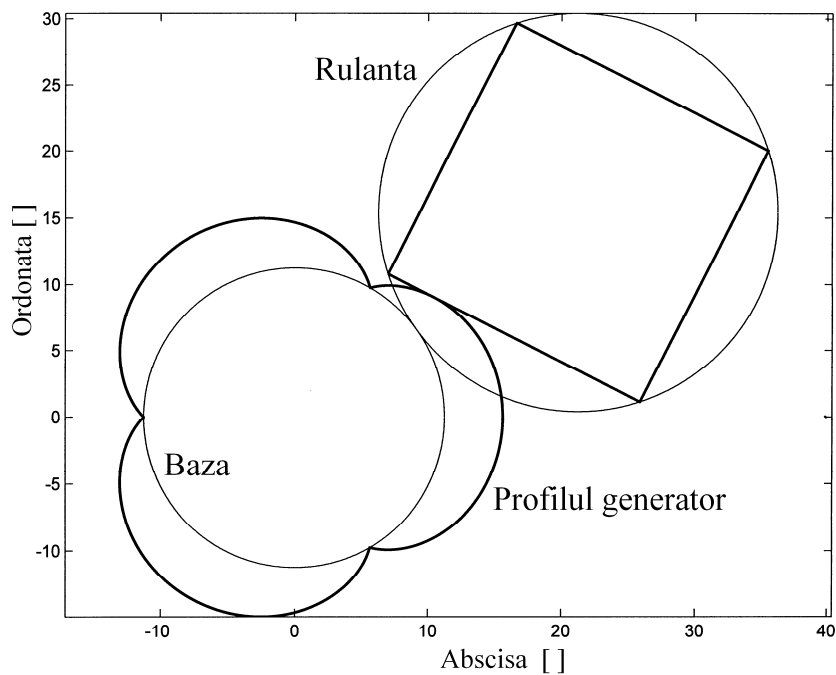


Figura 50. Redare grafică a schemei de generare a profilului generator cu trei lobi ($N_B=3$) pentru obținerea prin rulare a unui pătrat ($N_A=4$).

Ca exercițiu se pot genera profile generatoare cu orice număr de lobi schimbând corespunzător valoarea variabilei `nlob`.

La fel de bine se poate cerceta generarea profilului generator pentru orice curbă plană definită ca o succesiune de segmente (suficient de scurte pentru aproximarea arcelor elementare de curbă).

6. 2. Obținerea profilului de generat prin rularea profilului generator

Se poate face formal cel mai simplu dacă profilul generator se atașează rulantei iar rulanta rulează față de bază. Altfel spus, baza și rulanta se inversează între ele;

Relația (14) se rescrie ca:

$$(16) \quad N_A \cdot R_r = N_B \cdot R_b$$

Relația 15 se rescrie ca:

$$(17) \quad n_r/n_b = N_A/N_B$$

Înfășurătoarea pozițiilor succesive ale profilului generator este profilul de generat. Pe acest considerente s-a elaborat rogramul TRIUNGHI3 (redat mai jos și în Anexa 22) care permite definirea prin rulare a triunghiului echilateral avut în vedere anterior ca profil de generat.

Program TRIUNGHI3

```
clear all;close all;tic
%ACI INCEPE SINTEZA PROFILULUI DE SCULA PRIN RULARE
CERC PE CERC
nlob=1;%nlob este numarul de lobi ai profilului
generator
rrul=50;rbaz=rrul*nlob/3;xc=0;yc=0;
x(1)=-43.3013;y(1)=-25;
x(2)=43.3013;y(2)=-25;
x(3)=0;y(3)=50;
x(4)=-43.3013;y(4)=-25;
k=4;
%Sectiunea de definire a profilului de generat
incr=2*pi/(500);
alfa=0;beta=0;l=1;
for i=0:incr:2*pi;
```

```

    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+3)/3*i;
        x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*si
n(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;

```

```

        if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
%close all;plot(absc,ord,'k','LineWidth',1.5);axis
equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE PE
TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC PRIN
CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rbaz+rrul)/rrul*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*si
n(beta1);
        %x2(1)=x1(j);y2(1)=y1(j);
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',1);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;

```

```

%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',2);
l=1;
for i=0:.01:2*pi
xcerc(l)=(rbaz+rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)
=(rbaz+rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',2);
%RUTINA DE DETERMINARE A PROFILULUI piesei
%axis([-rbaz rbaz -rbaz rbaz+10])
toc

```

În figura 51 se prezintă rezultatul grafic al rulării programului pentru profil generator cu un singur lob ($N_B = 1$, $N_A = 3$).

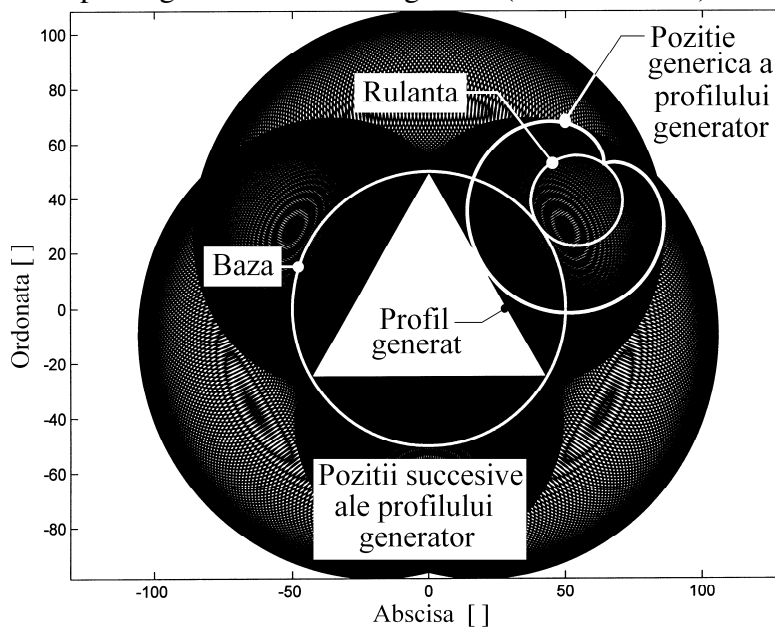


Figura 51. Rezultat grafic al execuției programului TRIUNGHI3, folosind profil generator cu un singur lob ($N_B=1$).

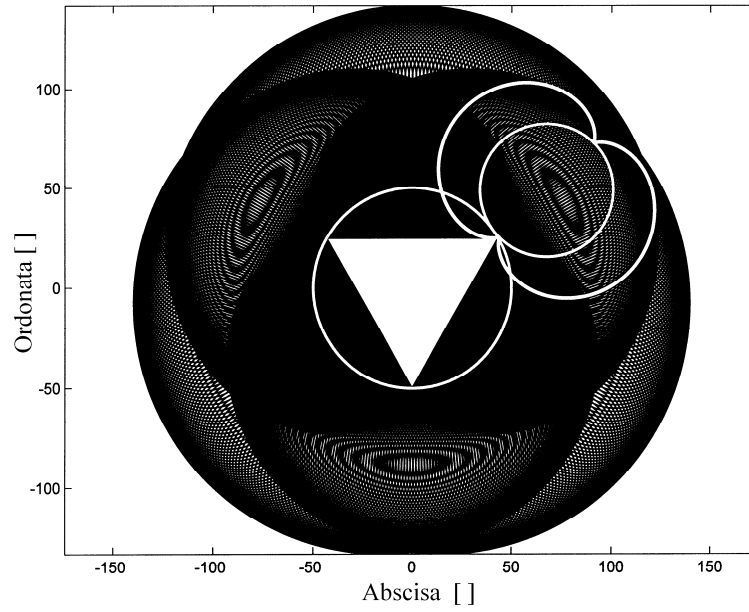


Figura 52. Rezultat execuție program TRIUNGHI3, folosind profil generator cu doi lobi ($N_B=2$).

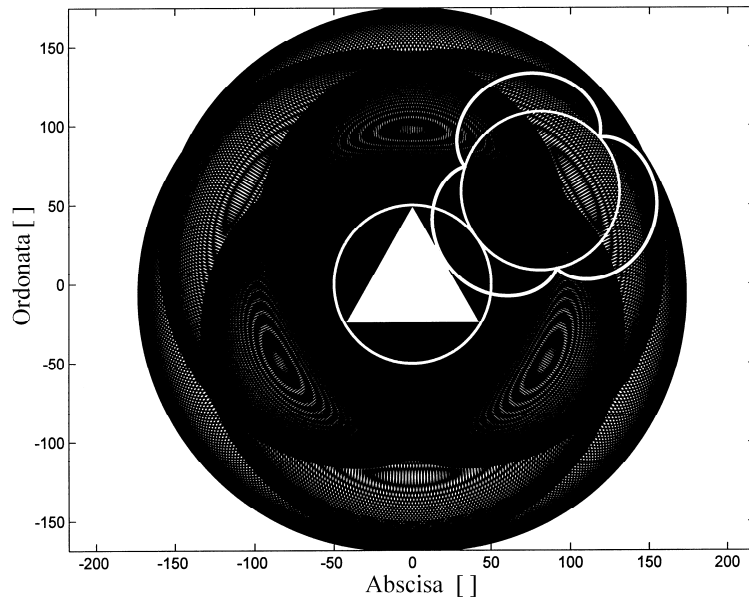


Figura 53. Rezultat execuție program TRIUNGHI3, folosind profil generator cu trei lobi ($N_B=3$).

În figura 52 se prezintă rezultatul grafic al rulării programului TRIUNGHI3 pentru profil generator cu un singur lob ($N_B = 2$, $N_A = 3$). În figura 53 se prezintă rezultatul grafic al rulării programului pentru profil generator cu un singur lob ($N_B = 3$, $N_A = 3$).

Dacă prima parte a programului TRIUNGHI3 (delimitată până la linia de comentariu **%Sectiunea de definire a profilului de generat**) se substituie cu secțiunea:

```
clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN RULARE
CERC PE CERC
nlob=3;%nlob este numarul de lobi ai sculei
(profilului generator)
a=10.6066;r=a*2^0.5;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-a;y(1)=-a;x(2)=a;y(2)=-a;
x(3)=a;y(3)=a;x(4)=-a;y(4)=a;
x(5)=-a;y(5)=-a;
k=5;
```

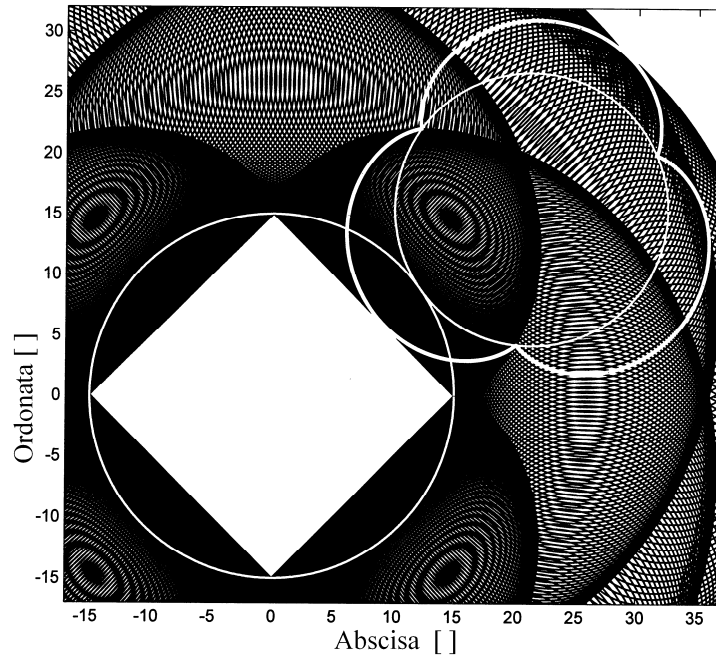


Figura 54. Rezultat generare pătrat folosind profil generator cu trei lobi ($N_B=3$).

și instrucțiunea:

```
alfa1=alfa+i;beta1=beta+(nlob+3)/3*i;
```

din primul ciclu `for-end` se substituie cu instrucțiunea:

```
alfa1=alfa+i;beta1=beta+(nlob+4)/4*i;
```

iar ca ultimă instrucțiunea se adaugă:

```
axis([-17 37 -17 32])
```

atunci se obține un program (care poate fi denumit PATRAT3, redat în Anexa 22 a) cu funcții identice cu TRIUNGHI3, destinat generării unui profil pătrat folosind o sculă cu trei lobi. Execuția acestuia conduce la generarea reprezentării grafice din figura 54.

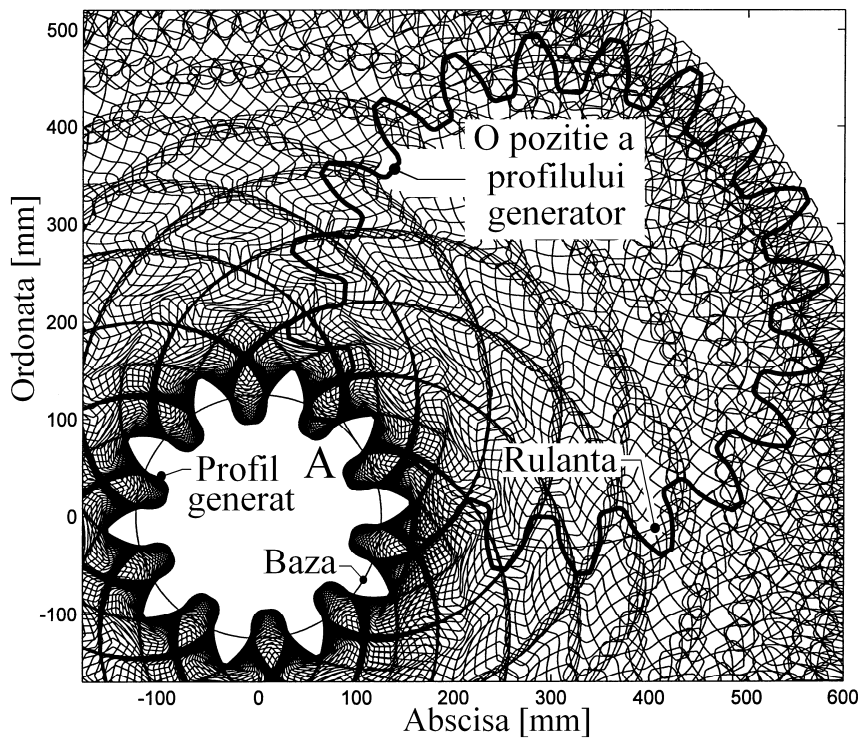


Figura 55. Rezultat generare a unei roți dințate prin rulare cerc pe cerc (programul SIMGENDANT1, Anexa 23)

Pe baza elementelor introduse anterior se poate acum cerceta generarea unui profil de roată dințată prin rularea cerc pe cerc. Propunem utilizarea ca profil generator profilul de roată dințată, descris în figura 6, obținut prin rularea programului DETPROF (cu listing-ul complet redat în Anexa 3). Programul DETPROF se copie într-un

folder separat. După execuția acestuia se tastează în fereastra de comandă Matlab și apoi se execută comenzile: `save absc;` `save ord;`

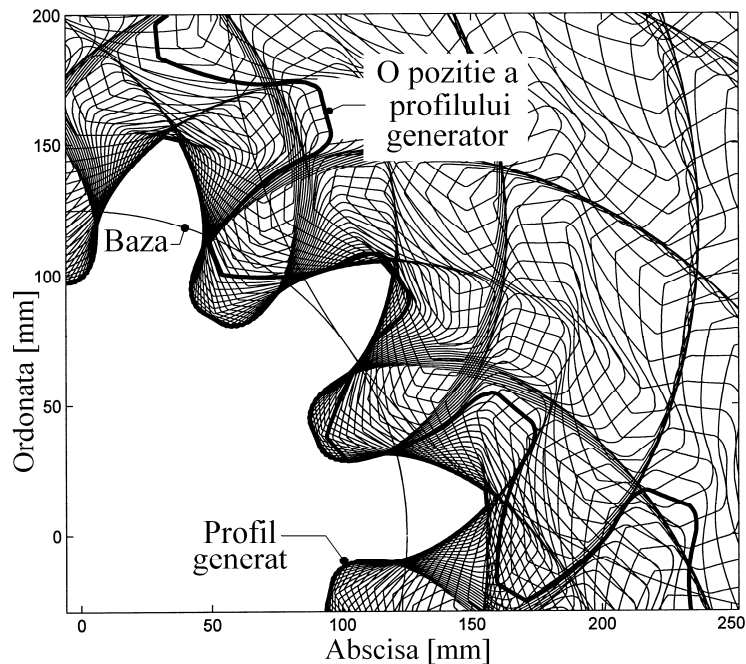


Figura 56. Detaliu grafic al figurii 55, în zona A.

Pe o structură foarte apropiată de programul TRIUNGHI3 se concepe programul de generare prin rulare cerc pe cerc a unei roți dințate cu zece dinți. Programul (numit SIMGENDANT1, cu listing-ul redat în Anexa 23) folosește profilul generator obținut anterior, încărcat la început cu instrucțiunile:

```
load absc; load ord;
```

Execuția acestui program conduce la rezultatul grafic din figura 55. S-a avut în vedere generarea unui profil de roată dințată cu 10 dinți (modulul este evident același cu al profilului generator). Pe figură sunt reprezentate pozițiile succesive ale profilului generator care definesc prin înfășurare profilul generat, cercul bază și cercul rulantă precum și o poziție generică a profilului generator (evidențiată cu altă grosime a liniei). Evident, cu cât numărul de poziții succesive ale profilului generator este mai mare cu atât profilul de generat este mai precis definit. Figura 56 prezintă un detaliu grafic al figurii 55 în zona A.

Dacă după rularea programului SIMGENDANT1 se introduce și se execută din fereastra de comandă secvența:

```
close all
plot(absc,ord,'k','LineWidth',1.5);axis equal;
hold on
plot(x1a,y1a,'k','LineWidth',1.5);l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);l=1;
for i=0:.01:2*pi
xcerc(l)=(rbaz+rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)
=(rbaz+rrul)*sin(alfa1)+rrul*sin(i);l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
axis equal;axis([-180 600 -170 520])
```

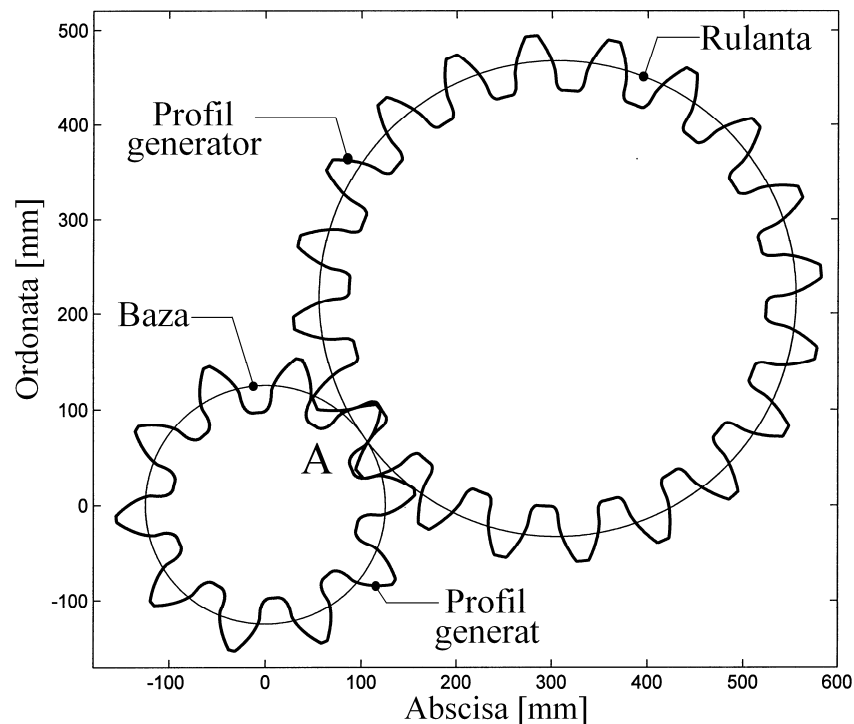


Figura 57. *Reprezentare complementară figurii 55. Reprezentare a profilelor (generator și generat) într-o poziția unică de rulare.*

atunci se obține reprezentarea profilului generat și a celui generator corespunzător unei poziții unice de rulare, conform figurii 57.

Pe figura 57 apare evident faptul că între cele două profile apare alunecare relativă pentru punctele de contact care nu se află în punctul de tangență al bazei cu rulanta. Această alunecare apare și în cazul angrenajelor și ea este responsabilă de uzura de contact a flancurilor active ale dinților roților dințate. Această alunecare are drept consecință apariția frecării între flancuri care afectează valoarea randamentului mecanic. Pentru diminuarea frecării este indicată ungerea transmisiilor cu roți dințate.

În practică, diametrul de cap al profilului generat rezultă din semifabricat (profilul generator, regăsit ca tăiș al cuțitului roată de mortezat nu este implicat în așchiere pentru efinirea zonei de cap al dinților pe roata de fabricat).

Daca se introduce în fereastra de comandă instrucțiunea: `axis([-30 275 -30 190])` atunci se obține o reprezentare din zona de contact al celor două profile (zona de „angrenare”) conform figurii 58.

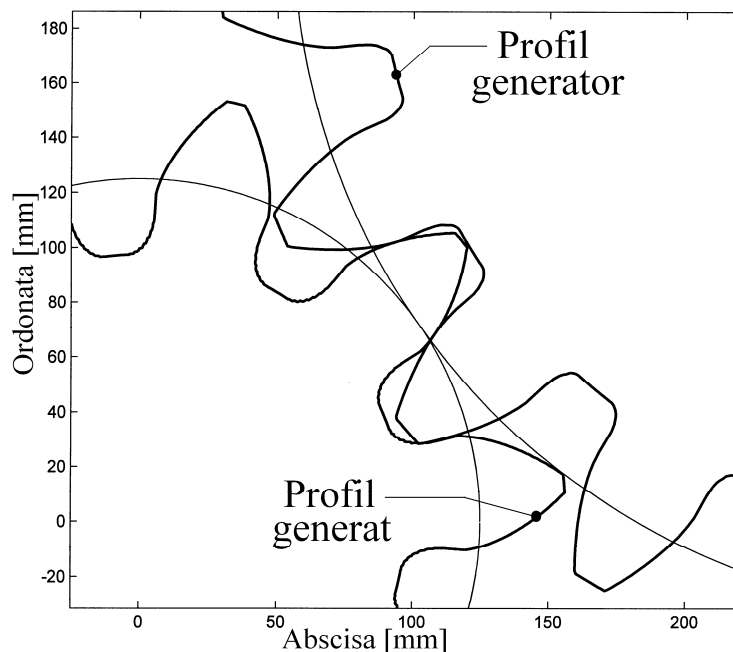


Figura 58. Detaliu grafic la figura 57, în zona A.

Reprezentarea din figura 58 poate servi foarte bine unor finalități cum ar fi:

- cercetarea caracteristicilor angrenării (gradul de acoperire, linia de angrenare);
- determinarea profilului real obținut prin rulare (având în vedere numărul limitat de poziții distincte ale profilului generator);
- înțelegerea aspectelor legate de contactul flanc-picior al dinților (aici subtăierea la baza dintelui pe profilul generat);
- cercetarea evoluției alunecării între flancurile dinților roților dințate.

O consecință importantă a rezultatului grafic din figura 58 este faptul că subtăierea la baza dintelui nu este neapărat un fenomen nedorit sau de evitat. Dacă admitem că în figura 58 se inversează profilul generator cu cel de generat, acesta se regăsește pe tăișul sculei cuțit roată de mortezat. De aici rezultă faptul că un profil cu subtăiere se regăsește transpus (prin rulare) pe piesă ca profil evolventic fără subtăiere. Și în acest caz se pot realiza studii cu referire la deplasarea danturii și efectele acesteia asupra subtăierii.

7. Aspecte ale simulării rulării cerc în cerc (profil generator pe sculă de tip arbore)

În afară de divizarea continuă în timpul procesului de fabricație un alt avantaj al mortezării cu cuțit roată este faptul că permite prelucrări care adesea sunt imposibile prin alte procedee. Un exemplu în acest sens este prelucrarea roților dințate cilindrice cu dantură evolventică interioară dreaptă sau elicoidală. În acest caz se folosește un cuțit roată de mortezat danturat exterior (sculă de tip arbore). Pentru acest procedeu generarea se face prin așa numita rulare cerc în cerc.

Mortezarea cu cuțit roată presupune ca rulanta și baza (ambele cercuri) să fie tangente interior. După cum profilul de generat (solidar cu baza) este interior respectiv exterior, rulanta este în exterior respectiv în interior (profilul generator rezultat se transpune pe cuțit de mortezat cu profil exterior respectiv interior).

O parte a considerentelor rulării geometrice cerc în cerc pot fi formulate pe baza reprezentării grafice din figura 59.

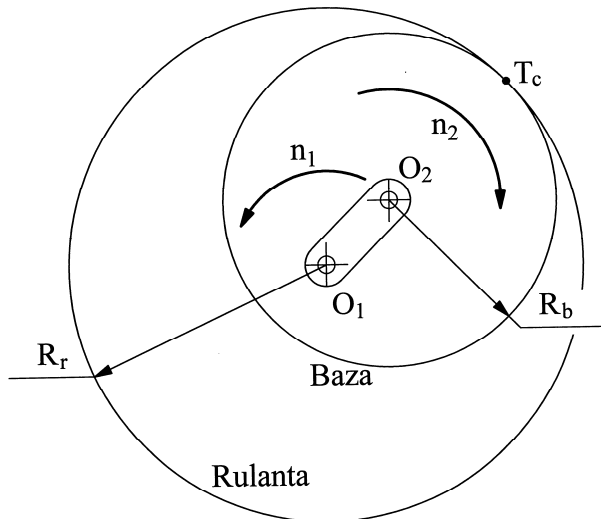


Figura 59. Elemente principale privind rularea cerc în cerc.

Cercul bază are raza R_b și are centrul în punctul O_2 . Cercul rulantă are raza R_r și centrul în punctul O_1 . Cele două cercuri sunt tangente interior în punctul T_c .

Să presupunem că rulantă este fixă iar rularea este efectuată de cercul de bază. Acesta trebuie să execute

simultan două mișcări de rotație:

- o mișcare de rotație cu turația n_1 a centrului O_2 în jurul centrului O_1 ;
- o mișcare de rotație cu turația n_2 a cercului rulantă în jurul propriului centru (O_2).

Cele două mișcări trebuie riguros corelate pentru a asigura rularea (viteză relativă este nulă în punctul T_c , acesta fiind centru instantaneu de rotație). Relația dintre cele două turații rezultă din ecuația vitezei tangențiale a punctului T_c (de pe bază), raportată la punctul O_1 .

$$(18) \quad v_{T_c} = v_{T_c O_2} + v_{O_2 O_1} + v_{O_1}$$

În (18) $v_{T_c} = 0$ deoarece T_c este centru instantaneu de rotație, $v_{T_c O_2} = 2 \cdot \pi \cdot n_2 \cdot R_b$ este viteza tangențială a punctului T_c față de punctul

O_2 , $v_{O_2O_1} = 2 \cdot \pi \cdot n_1 \cdot (R_r - R_b)$ este viteza tangențială a punctului O_2 față de punctul O_1 , $v_{O_1} = 0$ este viteza punctului O_1 (nulă, dacă considerăm acest punct fix). Cu aceste specificații relația (18) devine:

$$(19) \quad n_2 \cdot R_b = -n_1 \cdot (R_r - R_b) \quad \text{sau} \quad \frac{n_2}{n_1} = \frac{-(R_r - R_b)}{R_b}$$

Pe baza relației (19) se poate enunța condiția rulării cerc în cerc cu cercul rulantă fix și cercul bază mobil după cum urmează:

- dacă centrul O_2 al cercului bază se rotește în jurul centrului O_1 al cercului rulantă (considerat fix), cu un unghi α , atunci cercul bază se rotește în jurul propriului centru cu un unghi $\beta = -\alpha \cdot (R_r - R_b) / R_b$. Altfel spus, corelat cu (19), raportul celor două unghiuri este dat de relația:

$$(20) \quad \frac{\beta}{\alpha} = \frac{-(R_r - R_b)}{R_b}$$

Din rațiuni evidente, condiția rulării cerc în cerc cu ambele cercuri libere la rotație în jurul propriei axe (condiție folosită în fabricație) se enunță după cum urmează:

- dacă cercul bază se rotește în jurul propriei axe cu unghiul α , atunci cercul rulantă se rotește în jurul propriei axe cu unghiul $\beta = \alpha \cdot R_b / R_r$.

Să presupunem acum că, în condițiile de mai sus, un observator se plasează în centrul O_2 al cercului bază. Pentru observator centrul cercului rulantă (O_1) execută mișcare de rotație cu turația n_1 în jurul centrului O_2 , în același sens cu cel figurat în figura 59. Să presupunem că cercul rulantă execută mișcare de rotație în jurul centrului O_1 cu turația $n_3 = -n_2 \cdot R_b / R_r$. Ținând seama de faptul că cercul bază și cercul rulantă nu au alunecare relativă în punctul de tangență T_c , obligatoriu cercul bază trebuie să se rotească cu turația n' în jurul propriei axe care este în relație cu n_3 conform relației: $n' / n_3 = R_r / R_b$. Adică $n' = n_3 \cdot R_r / R_b = -n_2$. Cu aceasta turația rezultantă a roții bază (care se rotește cu turația

n_2) devine $n'_2 + n_2 = -n_2 + n_2 = 0$. Se obține astfel o nouă schemă de generare în care cercul bază este fix iar cercul rulantă execută mișcările n_1 și n_3 care pot fi puse în relație (ținând seama și de (19)) după cum urmează:

$$(21) \quad \frac{n_3}{n_1} = -\frac{n_2}{n_1} \cdot \frac{R_b}{R_r} \quad \text{sau} \quad \frac{n_3}{n_1} = \frac{(R_r - R_b)}{R_r}$$

Pe baza relației (21) se poate enunța condiția rulării cerc în cerc cu cercul bază fix și cercul rulantă mobil după cum urmează:

- dacă centrul O_1 al cercului rulantă se rotește în jurul centrului O_2 al cercului bază (considerat fix), cu un unghi α , atunci cercul rulantă se rotește în jurul propriului centru (O_1) cu un unghi $\beta = \alpha \cdot (R_r - R_b) / R_r$. Altfel spus, corelat cu (21), raportul celor două unghiuri este descris de relația:

$$(22) \quad \frac{\beta}{\alpha} = \frac{(R_r - R_b)}{R_r}$$

Condiția rulării cerc în cerc cu ambele cercuri libere la rotație în jurul propriei axe (condiție folosită în fabricație) rămâne cea enunțată anterior:

- dacă cercul bază se rotește în jurul propriei axe cu unghiul α , atunci cercul rulantă se rotește în jurul propriei axe cu unghiul $\beta = \alpha \cdot R_b / R_r$.

Cu acestea considerentele teoretice ale rulării cerc în cerc sunt statuate și se poate trece la utilizarea lor în simulare. Fie ca prim exemplu necesitatea simulării generării unui profil pătrat interior (pentru un alezaj). Și în acest caz se parcurg etapele următoare (la care s-a făcut deja referire anterior):

1. Determinarea profilului generator prin simulare. Acesta este regăsit pe sculă, cuțit roată de mortezat. Pentru obținerea grafică și analitică a lui, în condițiile figurii 59, se atașează cercului

rulantă profilul teoretic de generat (pătrat înscris în cercul rulantă). Cercul rulantă rulează peste cercul bază în condițiile relațiilor (21), (22) și generează într-un plan solidar cu cercul bază o succesiune de poziții ale pătratului. Aceste poziții înfășoară **la interior** profilul generator. Razele cercurilor bază și rulantă trebuie să satisfacă relația:

$$(23) \quad N_b \cdot R_b = N_r \cdot R_r \quad \text{sau} \quad R_b = N_r \cdot R_r / N_b$$

În (23) N_b și N_r sunt numere întregi. Pentru profile de generat sub forma unui poligon (obligatoriu regulat), N_b reprezintă numărul de laturi (aici, pentru un pătrat, $N_b = 4$). Pentru profile regulate de altă formă (roată dințată, arbore canelat, etc.) N_b este egal cu numărul de dinți (caneluri). Profilul generator va avea o formă care depinde de profilul de generat dar și de N_r . În (23) $R_b < R_r$ deci $N_r < N_b$. Din rațiuni ce vor rezulta ulterior se prezumă în plus că $N_r > 1$.

Maniera de determinare a profilului generator (definit prin segmente de dreaptă) a fost deja prezentată și utilizată anterior (de exemplu în orizontul figurii 6 cu detaliere în programul DETPROF din Anexa 3).

2. Simularea obținerii profilului de generat prin rulare. Se folosește schema din figura 59 considerând că cercul rulantă este fix și cercul bază rulează în interiorul acestuia, în condițiile din relațiile (19) și (20). Profilul generator (obținut anterior, solidar cu cercul bază) execută mișcările n_1 și n_2 . Înfășurătoarea **la exterior** a pozițiilor succesive ale profilului generator într-un plan solidar cu cercul bază este profilul de generat. Acesta trebuie să fie evident pătratul folosit inițial, înscris în cercul de bază.

Programul **fig60** (Anexa 24) realizează simularea profilului generator prin rularea cerc în cerc a profilului de generat. Execuția acestuia conduce la reprezentarea din figura 60. Sunt reprezentate

aici pozițiile succesive ale profilului de generat (pătrat), o poziție generică a acestui pătrat precum și profilul generator obținut ca înfășurătoare a pozițiilor succesive ale profilului de generat.

Programul fig60

```
clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI GENERATOR PRIN
RULARE CERC IN CERC
nlob=3;%nlob este numarul de lobi ai profilului
%generator,  $N_r$  în relația (23);
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
%plot(x,y,'k','LineWidth',1.5);hold on
k=5;
incr=2*pi/(50);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',0.5);
hold on;axis equal;l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI GENERATOR
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end
```



```

end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end
end;l=l+1;
end
plot(absc,ord,'k','LineWidth',2.5);axis equal;
k=5;l=1;clear x1;clear y1;
for i=0.0:incr:0.0+incr;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-rbaz)/rrul*i;

```

```

        x1(j)=(-1)*(rbaz-
rrul)*cos(alfal)+x(j)*cos(beta1)-y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfal)+y(j)*cos(beta1)+x(j)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
plot(x1,y1,'k','LineWidth',2.5)

```

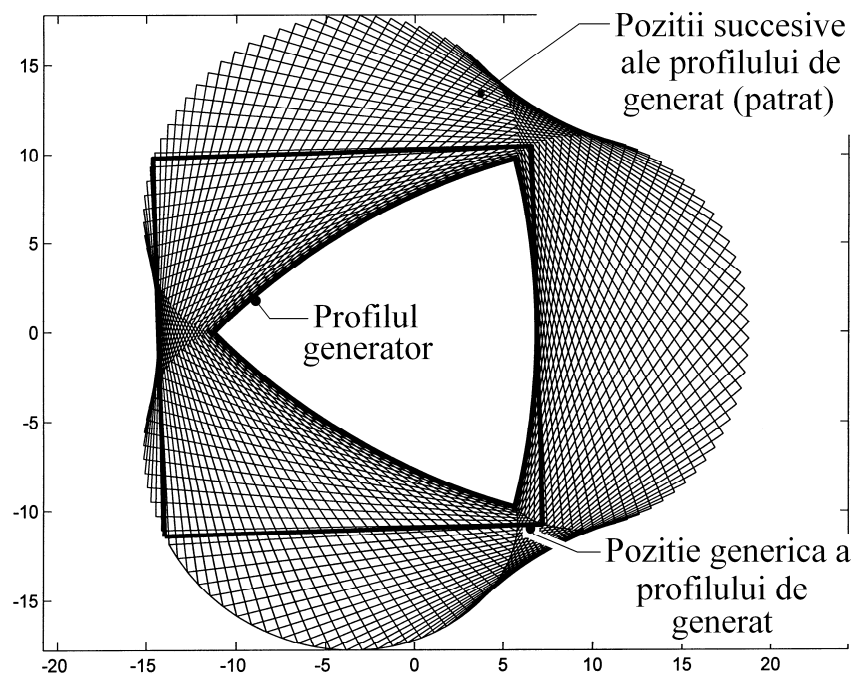


Figura 60. Ilustrare grafică a obținerii profilului generator cu $N_r=3$ și $N_b=4$ (execuție a programului `fig60`, redat în Anexa 24).

Se observă faptul că unui profil de generat cu patru laturi ($N_b=4$) îi corespunde un profil generator cu trei ($N_r=3$) arce de curbă (cicloide).

Dacă în programul `fig60` se blochează la execuție instrucțiunea marcată cu caractere îngroșate și subliniată și se adaugă setul de instrucțiuni:

```

l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;

```

```

end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rrul-
rbaz)*cos(alfal)+rrul*cos(i);ycerc(l)=(rrul-
rbaz)*sin(alfal)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);axis equal;

```

se obține la execuție reprezentarea grafică din figura 61.

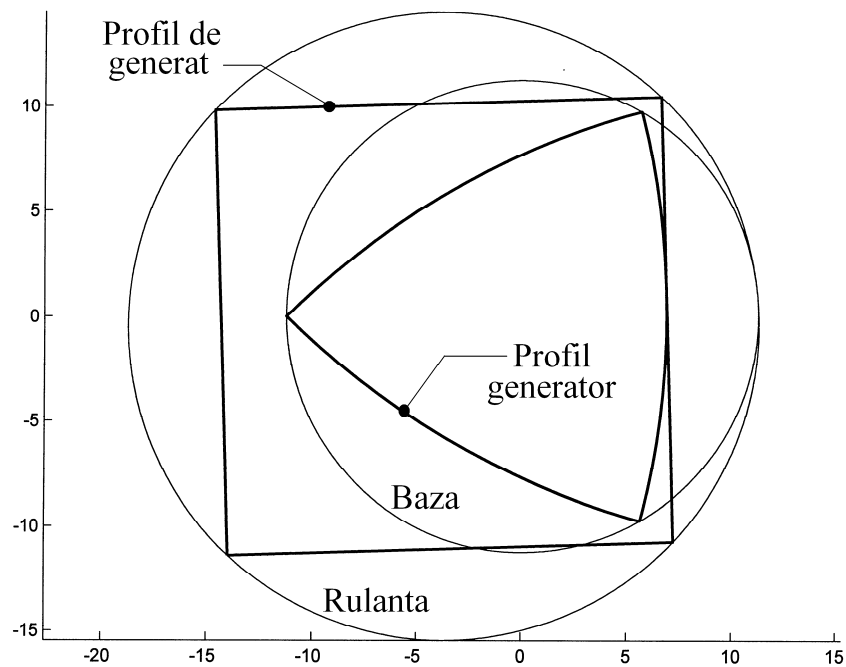


Figura 61. Sinteza a reprezentării din figura 60 (execuție a programului fig61, Anexa 24 a). O poziție generică a rulării (profilul de generat, profilul generator, cercurile bază și rulantă).

Pe figura 61 se observă faptul că din condiția ca profilul de generat să fie înscris în cercul rulantă rezultă că profilul generator este înscris în cercul bază.

Profilul generator obținut în prima etapă poate fi utilizat pentru obținerea prin rulare cerc în cerc a profilului de generat. Se consideră de această dată cercul rulantă fix, profilul generator solidar cu cercul bază care este mobil (execută mișcările n_1 și n_2 din figura 59). Rularea cercului bază (care se comportă ca rulantă) în interiorul cercului rulantă (care se comportă ca bază) se produce în condițiile relațiilor (19) și (20). Într-un plan solidar cu cercul rulantă se generează pozițiile succesive ale profilului generator a căror înfășurătoare (la exterior) trebuie să fie profilul de generat (în cazul nostru un pătrat).

Pe aceste considerente s-a elaborat programul fig62 (redat complet în Anexa 25), care completează programul fig60. Programul simulează rularea incompletă a profilului generator, cu rezultat grafic redat în figura 62.

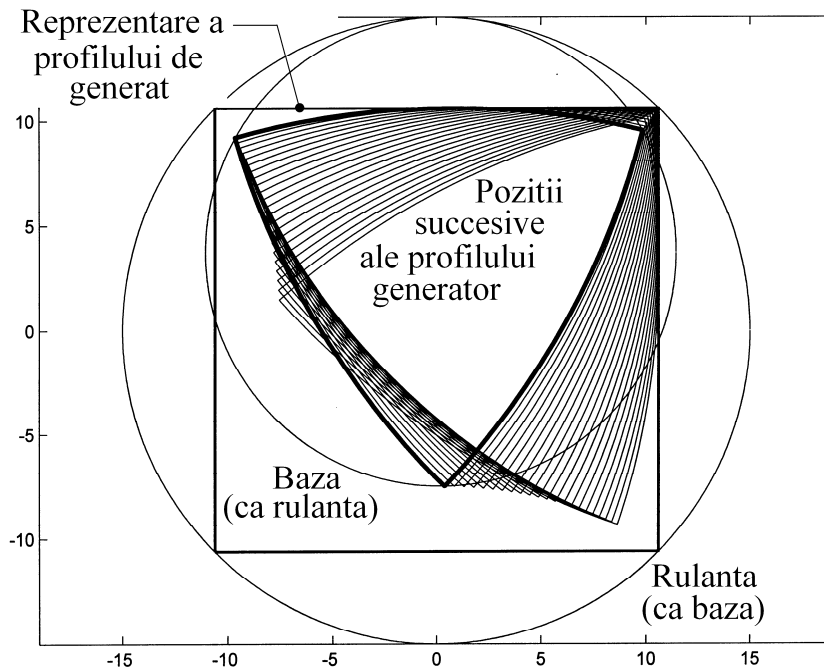


Figura 62. Simulare parțială a obținerii profilului de generat prin rulare cerc în cerc (execuția a programului fig62 redat în Anexa 25).

Se simulează generarea unui colț al pătratului și câteva poziții succesive ale profilului generator. Simularea rulării complete, cu redare

grafică a rezultatului (în figura 63), se realizează prin execuția programului PATRAT2 (redat în Anexa 26).

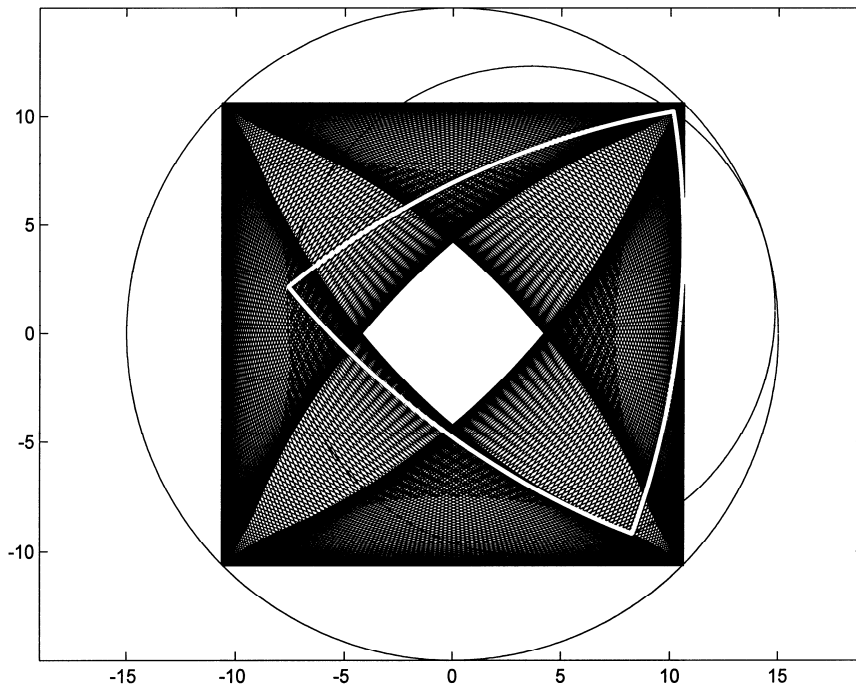


Figura 63. Simulare completă a obținerii profilului de generat prin rulare cerc în cerc (cu $N_r=3$ și $N_b=4$). Rezultat grafic al execuției programului PATRAT2.

Pe figură sunt redată toate pozițiile profilului generator, înfășurate la exterior de către profilul de generat (pătrat), o poziție a profilului generator (marcat cu alb).

Utilitatea practică a acestei simulări este legată de obținerea alezajelor cu profil transversal pătrat prin rulare, utilizând procesul de mortezare cu cuțit roată, pe care se regăsește materializat ca tăiș profilul generator. Invităm cititorul să reflecteze asupra unei probleme de tehnologie: în semifabricat trebuie practicat un alezaj inițial în care să intre scula la debutul procesului.

Se poate testa acum rularea cerc în cerc pentru generarea pătratului, cu parametrul $N_r = 2$ (în programul PATRAT2, în linia 2 se

înscrie nlob=2;

Rezultatul grafic este reprodus în figura 64, în aceiași termeni cu cei deja utilizați în figura 63 (profilul generator reprezentat cu alb).

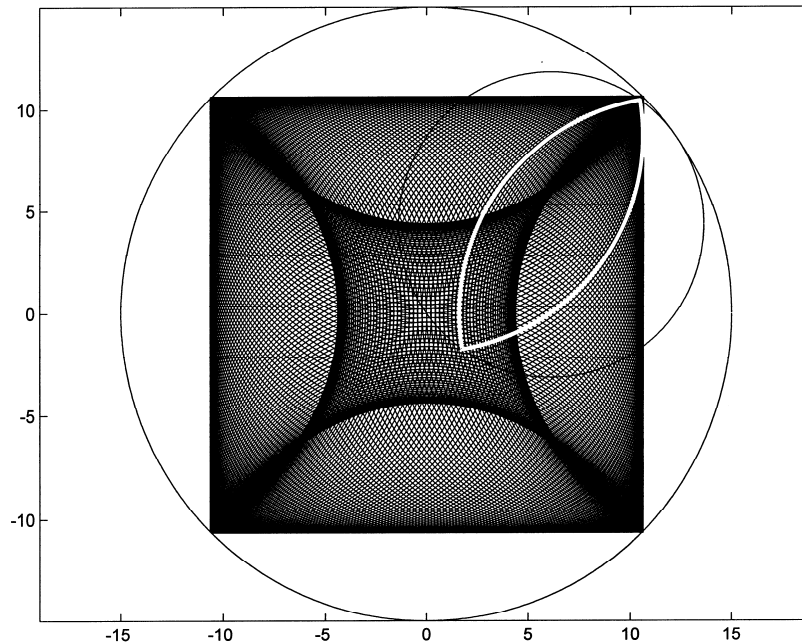


Figura 64. Simulare completă a obținerii profilului de generat prin rulare cerc în cerc (cu $N_r=2$ și $N_b=4$).

Deși rezultatul pare utopic din punct de vedere tehnologic, profilul generator din figura 64 poate fi mai ușor utilizat, plecând de la un alezaj preliminar prelucrat în piesă, în care să încapă profilul generator la debutul fabricației prin rulare-mortezare.

Evident, se poate simula rularea cerc în cerc pentru orice profil de generat. În figura 65 se prezintă rezultatul grafic al simulării generării prin rulare a unui pentagon regulat ($N_r=3$, $N_b=5$). Programul aferent se obține dacă secțiunea de început a programului PATRAT2 (până la instrucțiunea din linia 11, $k=5$;) se înlocuiește cu secvența:

```
clear all;close all;tic
nlob=3;
r=6.18;a=100;b=100;
```

```

rrul=r;rbaz=rrul*nlob/5;xc=0;yc=0;
x(1)=-5.8779;y(1)=1.9098;x(2)=-3.6327;
y(2)=-5;x(3)=3.6327;y(3)=-5;
x(4)=5.8779;y(4)=1.9098;x(5)=0;y(5)=6.1803;
x(6)=-5.8779;y(6)=1.9098;k=6;

```

Pentagonul este descris prin coordonatele celor cinci segmente care definesc laturile. Rularea programului în aceste condiții duce la rezultatul grafic din figura 65.

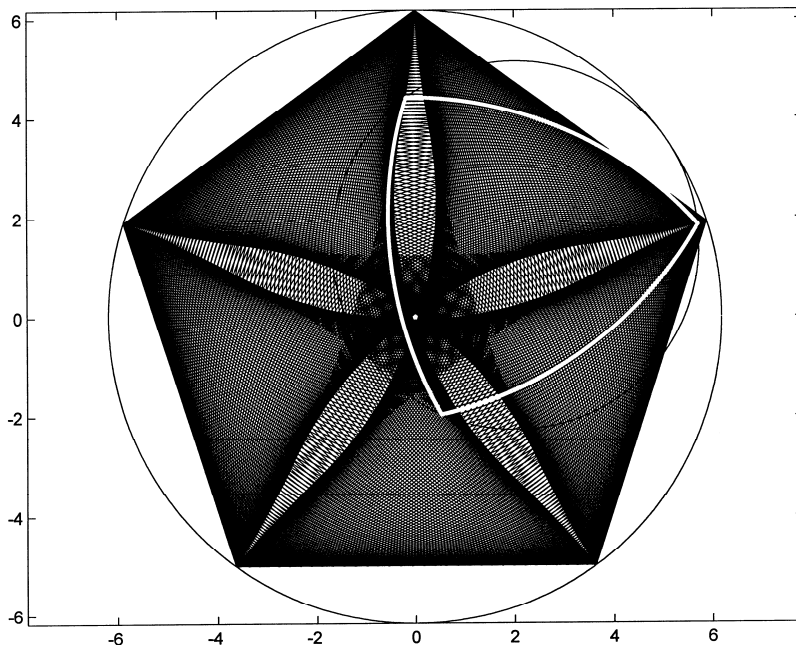


Figura 65. Simulare a obținerii prin rulare cerc în cerc a unui pentagon regulat (cu $N_r=3$ și $N_b=5$).

Sunt reprezentate cercurile bază și rulantă, o poziție a profilului de generat (cu culoare albă) precum și toate pozițiile acestuia care definesc prin înfășurare la exterior profilul de generat. Și aici se oferă un exemplu de obținere prin mortezare cu cuțit roată (cu profilul generator ca tăiș) pentru un alezaj. Piesa conjugată, de tip arbore, se poate obține de exemplu prin rulare cerc-cerc (utilizând programul PENTAGON1 din Anexa 7).

Se poate simula și obținerea prin rulare a unui pentagon în condițiile $N_r=4$ și $N_b=5$ prescriind în program valoarea:

$nlob=4$; .Rezultatul grafic este redat în figura 66 în termeni deja prezentați anterior.

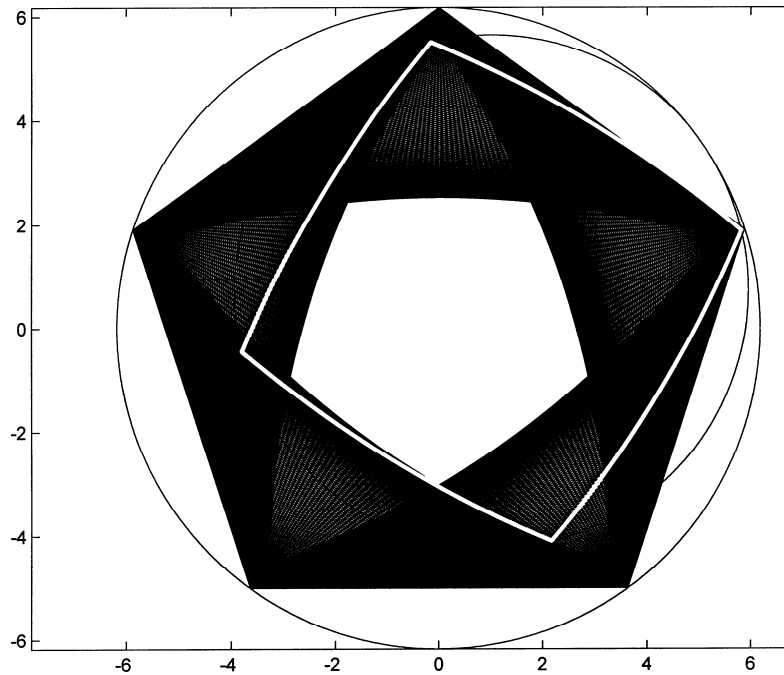


Figura 66. Simulare a obținerii prin rulare cerc în cerc a unui pentagon regulat (cu $N_r=4$ și $N_b=5$).

Asemănător, se poate cerceta obținerea prin rulare cerc în cerc a unui profil interior, de tip alezaj cu secțiune canelată ($N_r=4$ și $N_b=6$), cu șase caneluri, ca profil teoretic conjugat unui profil identic, dar pe arbore, deja discutat în orizontul figurii 23. Pentru aceasta, se utilizează programul PATRAT2 a cărui primă secțiune (până la instrucțiunea din linia 11, $k=5$;) se înlocuiește cu ciclul de instrucțiuni):

```
clear all;close all;tic;nlob=4;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob/6;
xc=0;yc=0;incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-incdef:78.7789*cons;
x(k)=12*cos(i);
y(k)=12*sin(i);
k=k+1;
```



```

end;
k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-incdef:68*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-
incdef:52*cons;x(k)=10*cos(i);y(k)=10*sin(i);
k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=120*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis equal
k=321;l=1;

```

Execuția separată a acestei secțiuni conduce la reprezentarea grafică a profilului de generat din figura 19. b. (aici elementul de intrare în procedeul de sinteză a profilului generator regăsit pe sculă).

Prin substituția-adiția secvenței de mai sus la programul PATRAT1 se obține programul CANELURAM3 cu listing-ul redat în Anexa 27. Rularea acestuia conduce la rezultatul grafic redat în figura 67. Profilul generator (transpus pe o sculă de tip arbore) are patru caneluri corespunzător valorii variabilei $N_r = 4$.

Se impune precizare că acesta este un demers pur teoretic. Profilul generat astfel nu este conjugat celui regăsit pe un eventual

arbore canelat (abordat în figura 23) din cauza creștăturilor dintre flancuri și cercul de picior care trebuie evitate aici. Din punct de vedere teoretic interesează aici corectitudinea obținerii flancurilor și a teșirilor (rectilinii) și a zonelor de cap și de picior (circular).

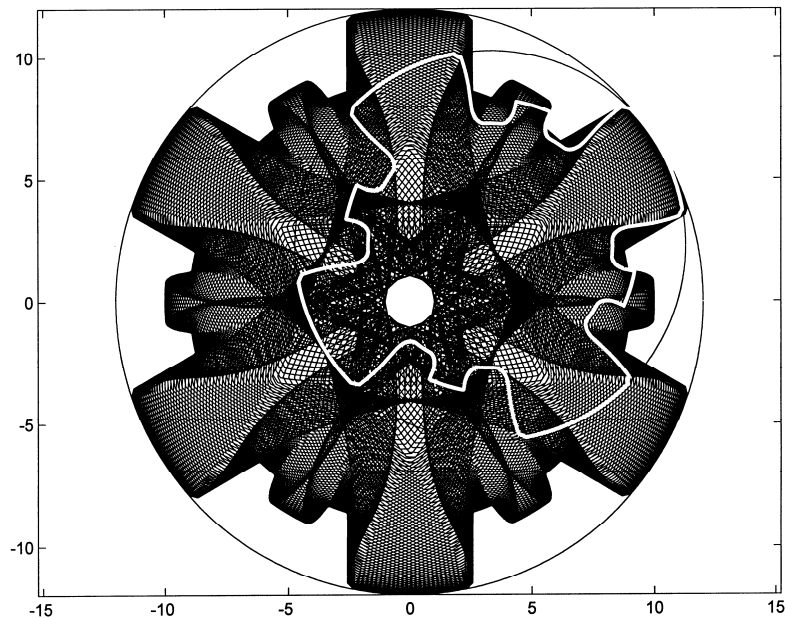


Figura 67. Simulare a obținerii prin rulare cerc în cerc a unui profil de tip alezaj canelat (cu $N_r=4$ și $N_b=6$).

Pentru a obține profilul conjugat celui din figura 23 ar trebui considerat că profilul de generat este cel deja descris în figura 19. a. Programul CANELURAM4 (cu listing-ul redat în anexa 28) asigură generarea profilului de alezaj conjugat profilului de arbore canelat din figura 23, cu redare grafică a rezultatului execuției în figura 68.

Din reprezentarea din figura 68 rezultă că prelucrarea alezajului canelat este posibilă tehnologic prin practicarea unei găuri în semifabricat, cu diametrul apropiat de diametrul interior al canelurii.

În figura 69 se prezintă un detaliu al figurii 68, corespunzător zonei A, care poate fi asociat detaliului de arbore canelat din figura 24 (ca piese conjugate). Se observă acuratețea descrierii profilului,

inclusiv un viciu de rulare foarte redus (profil de trecere în zona de racordare flanc-picior).

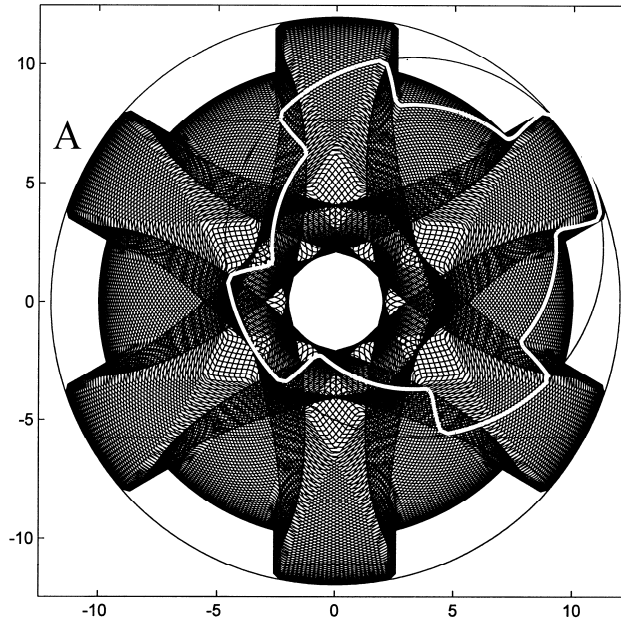


Figura 68. Simulare a obținerii prin rulare cerc în cerc a profilului de alezaj canelat conjugat profilului de arbore din figura 23.

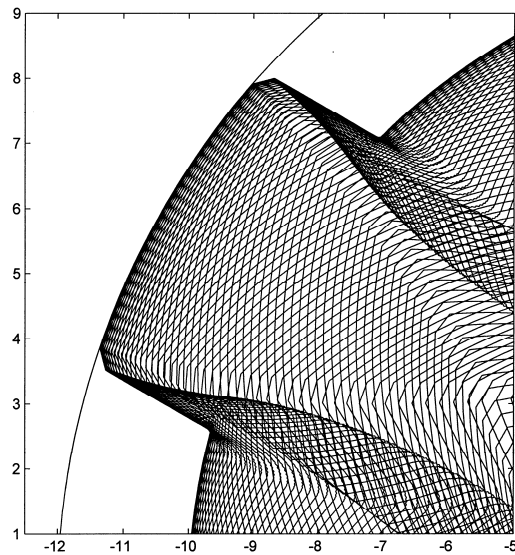


Figura 69. Detaliu grafic al figurii 68, în zona A.

Desigur cea mai completă valorificare a acestui tip de generare ar fi simularea rulării cerc în cerc cu sculă de tip alezaj pentru un profil regăsit pe o roată dințată danturată interior (mortezare cu cuțit roată danturat exterior).

Să considerăm necesitatea generării unui profil de roată dințată danturată interior cu 30 de dinți, așa cum rezultă din rularea programului DETPROF (redat în Anexa 3). Se copie programul într-un folder separat, se modifică numărul de dinți prin instrucțiunea $z=30$; din linia 2 și se comandă execuția. După execuție se introduc în fereastra de comandă și se execută instrucțiunile: `save absc; save ord`; Prin aceasta profilul roții dințate de generat (30 dinți, modul 25 mm, diametru de divizare 375 mm) este salvat.

În principiu, dacă prima secțiune a programului PATRAT2 (până la instrucțiunea din linia 11, $k=5$;) se elimină și se înlocuiește cu setul de instrucțiuni:

```
clear all;close all;tic
nlob=21;
rrul=375;rbaz=rrul*nlob/30;
xc=0;yc=0;
load absc;load ord;
clear x1;clear y1;
x=absc;y=ord;
xroata=x;yroata=y;
dim1=size(y);dim=dim1(2);
x(dim+1)=1000;y(dim+1)=0;dim=dim+1;
clear absc;clear ord
k=dim;
```

atunci se obține programul de simulare pentru obținerea prin rulare cerc în cerc a profilului generator (cu 21 dinți, regăsit pe sculă, conform instrucțiunii din linia a doua) și de generat (cu 30 dinți, regăsit prin fabricație pe piesă). Listing-ul complet al acestui program (SIMGENDANT2) se regăsește în Anexa 29.

Execuția acestui program conduce la rezultatul grafic din figura 70, în termeni deja consacrați anterior. Acuratețea descrierii celor două profile (de generat și generator) precum și relația dintre acestea rezultă

din reprezentarea din figura 71 (un detaliu în zona A la figura 70).

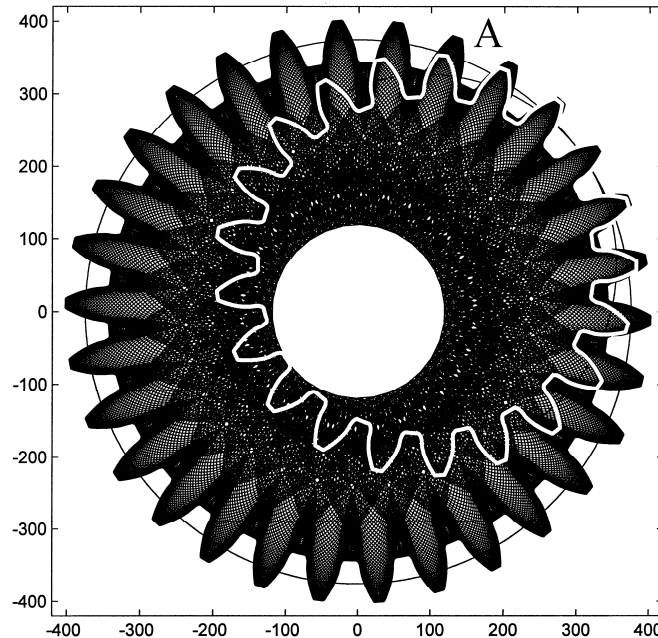


Figura 70. Simulare a obținerii prin rulare cerc în cerc a unui profil de roată dințată danturată interior (cu $N_r=21$ și $N_b=30$).

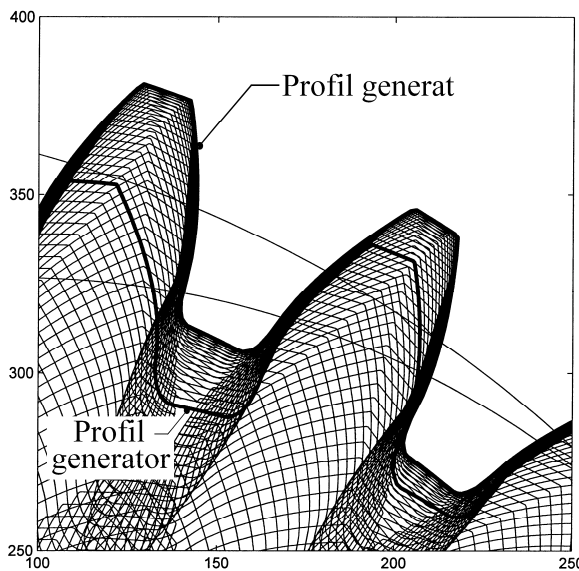


Figura 71. Detaliu în zona A la figura 70.

Reprezentarea din figura 70 s-a obținut introducând și executând instrucțiunile:

```
plot(x1,y1,'k','LineWidth',2.5);
axis([100 250
250 400]);
plot(xroata,yroata,'k','LineWidth',2.5);
```

Spre deosebire de figura 70, aici o poziție generică a

profilului generator care este reprezentată cu linie neagră, groasă. Pe figura 71 s-a făcut și reprezentarea profilului generat. Acesta se confundă practic cu înfășurarea la exterior a pozițiilor succesive ale profilului generator.

Dacă după execuția programului SIMGENDANT2 se introduce și se execută din fereastra de comandă instrucțiunile:

```
close all
plot(x1,y1,'k','LineWidth',1.5);axis equal;hold on;
plot(xroata,yroata,'k','LineWidth',1.5);
```

atunci se obține reprezentarea grafică din figura 72. Aici se reprezintă o

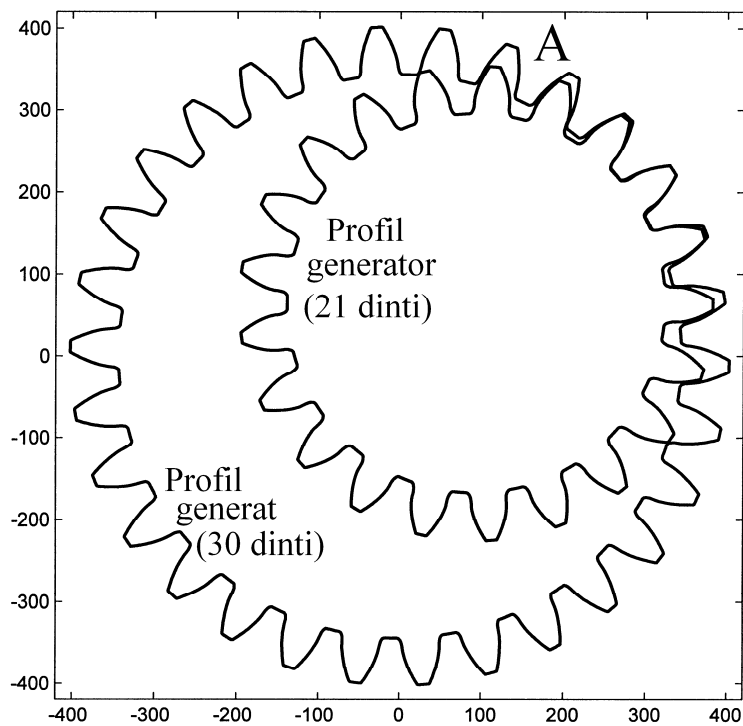


Figura 72. *Reprezentare grafică a profilului generat. O poziție a profilului generator în cursul rulării.*

poziție a profilului generator în decursul rulării precum și profilul generat. Se face precizarea că reprezentarea din figura 72 nu descrie

neapărat un angrenaj interior ci o sculă în poziție relativă față de piesă. Desigur, profilul generator și generat se pot asimila ca aparținând unor roți dințate în angrenare, de regulă însă, având în vedere faptul că fiecare roată este generată prin rulare pe scheme de rulare (și cu scule) diferite acestea nu sunt perfect conjugate în zona de picior roată danturată exterior. Funcționarea corectă presupune existența jocului la fund. Din acest motiv se corijează diametrele de cap (care de fapt rezultă din semifabricat și nu prin rulare). Se crește diametrul de cap pentru roata danturată interior și se scade acest diametru pentru roata danturată exterior.

Dacă se execută instrucțiunea `axis([100 350 150 400]);` se obține reprezentarea din figura 73, ca detaliu în zona A la figura 72.

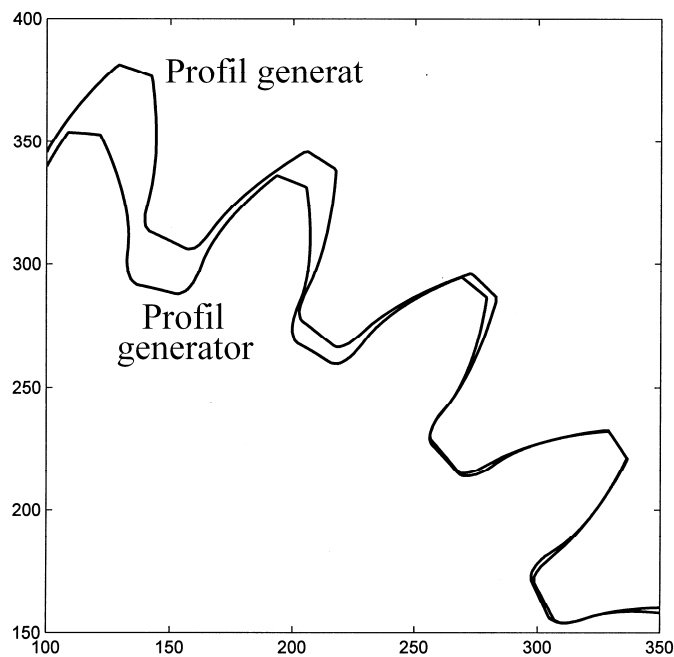


Figura 73. *Reprezentare a figurii 72 în zona detaliului A.*

Cel puțin acest tip de rulare prezintă importanță practică imediată având în vedere fabricația prin mortezare a roților dințate cu dantură interioară. Cititorul poate utiliza cele două programe

(DETPROF și SIMGENDANT3) pentru diverse configurații legate de valorile parametrilor N_r și N_b .

Se pot exercisa o multitudine de variante de simulare în legătură cu rularea cerc în cerc cu sculă de tip arbore pentru diferite tipuri de profile (roată de lanț, roată de clichet, alte profile neconvenționale, regulate și neregulate)

8. Aspecte ale simulării rulării cerc în cerc (profil generator pe sculă de tip alezaj)

În § 7 s-a abordat problema simulării rulării cerc în cerc care în fabricație se traduce în piesă de tip alezaj prelucrată cu sculă de tip arbore (de exemplu mortezare cu cuțit roată a unei roți dințate danturate interior). Din punct de vedere teoretic este la fel de posibilă abordarea rulării cerc în cerc cu fabricația prin mortezare a unei piese de tip arbore cu sculă de tip alezaj (de exemplu mortezare a unei roți dințate danturate exterior cu un cuțit roată danturat interior). Chiar dacă acest tip de generare nu este întâlnit în practică, există cel puțin rațiunea curiozității pentru simularea rulării de acest tip.

Este interesant de remarcat faptul că majoritatea considerațiilor teoretice din § 7 rămân valabile, situația de aici fiind o extindere a lor. Mai întâi, figura 59 trebuie privită ca având cercul bază în relație cu cercul rulantă conform cu: $R_b > R_r$. Relația (23) rămâne pe deplin valabilă, numerele N_r , N_b sunt întregi dar $N_r > N_b$.

Rămân formal valabile etapele de generare enunțate anterior la § 7. Elementele de specificitate sunt marcate mai jos:

1. Determinarea profilului generator prin simulare. Acest va fi regăsit pe sculă, cuțit roată de mortezat de tip alezaj. Pentru obținerea grafică și analitică a lui, în condițiile figurii 59 (cu $R_b > R_r$), se atașează cercului rulantă profilul teoretic de generat. Cercul rulantă rulează peste cercul bază în condițiile relațiilor (21), (22) și generează într-un plan solidar cu cercul bază o succesiune de poziții ale profilului teoretic de generat.

Înfășurătoarea **la exterior** a acestor poziții este profilul generator. În § 7 se utilizează înfășurătoarea la interior.

2. Simularea obținerii profilului de generat prin rulare. Se folosește schema din figura 59 considerând că cercul rulantă este fix și cercul bază rulează în exteriorul acestuia, în condițiile din relațiile (19) și (20). Profilul generator (obținut anterior, solidar cu cercul bază) execută mișcările n_1 și n_2 . Înfășurătoarea **la interior** a pozițiilor succesive ale profilului generator într-un plan solidar cu cercul bază este profilul de generat. În § 7 se utilizează înfășurătoarea la exterior.

Elementele de programare utilizate anterior în programul PATRAT2 pot fi utilizate aproape în întregime. Excepție face procedura de determinare a profilului generator. În secțiunea corespunzătoare se înlocuiește instrucțiunea `mindist=1000`; cu instrucțiunea: `mindist=-1000`. Toate instrucțiunile de forma: `if dist<mindist` se înlocuiesc cu instrucțiuni: `if dist>mindist`.

Folosind, cu titlu de exemplu, un profil necesar a fi generat de forma unui pătrat (deja utilizat anterior, cu $N_b = 4$), obținerea profilului generator prin rulare conform etapei 1 (cu $N_r = 5$) se poate ilustra grafic în figura 74 (rulare a programului `fig74`, redat în Anexa 30). Înfășurătoarea **la exterior** a pozițiilor profilului de generat este profilul generator.

Simularea obținerii profilului de generat prin rulare (conform etapei 2, cu utilizarea profilului generator deja introdus în figura 74) se poate face rulând programul PATRAT3 (redat în Anexa 31), cu rezultat grafic prezentat în figura 75 pe care sunt reliefate următoarele:

- o poziție generică a profilului generator (cu alb) și toate pozițiile acestuia pe parcursul rulării (cu negru);
- o poziție a cercului bază în care este înscris profilul generator (reprezentat cu alb, utilizat aici ca rulantă);
- o poziție a cercului rulantă în care este înscris profilul de generat (reprezentat cu alb, utilizat aici ca bază).

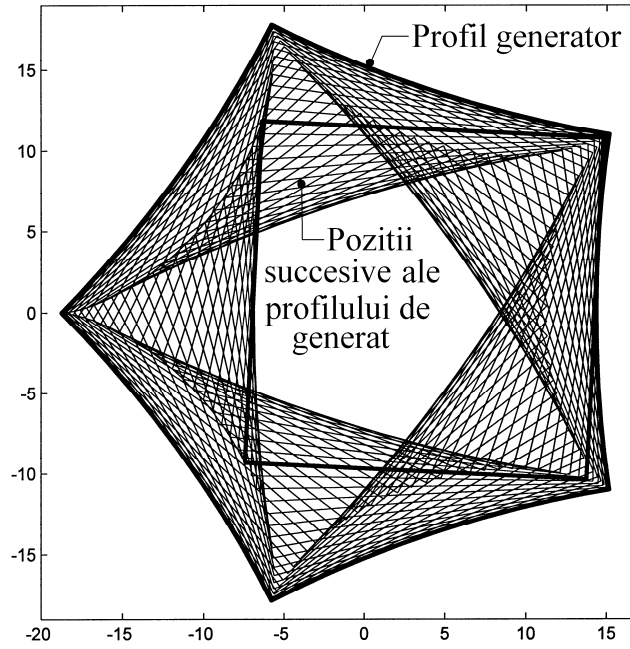


Figura 74. *Ilustrativă la obținerea profilului generator.*

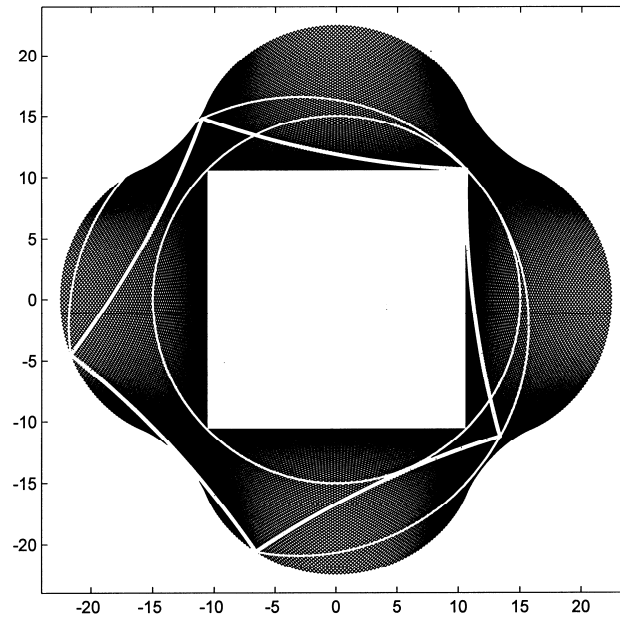


Figura 75. *Ilustrativă la obținerea profilului de generat. Figură generată de programul PATRAT3.*

Pe figura 75 este evident faptul că înfășurătoarea la interior a pozițiilor successive ale profilului generator este profilul de generat, aici pătratul de la care s-a plecat.

Dacă, după execuția programului PATRAT3, se execută setul de instrucțiuni:

```
close all
plot(x1,y1,'k','LineWidth',1.5);
axis equal;hold on;
plot(xroata,yroata,'k','LineWidth',1.5);
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);
    ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-rrul)*cos(alfal)+rrul*cos(i);
    ycerc(l)=(rbaz-rrul)*sin(alfal)+rrul*sin(i);
```

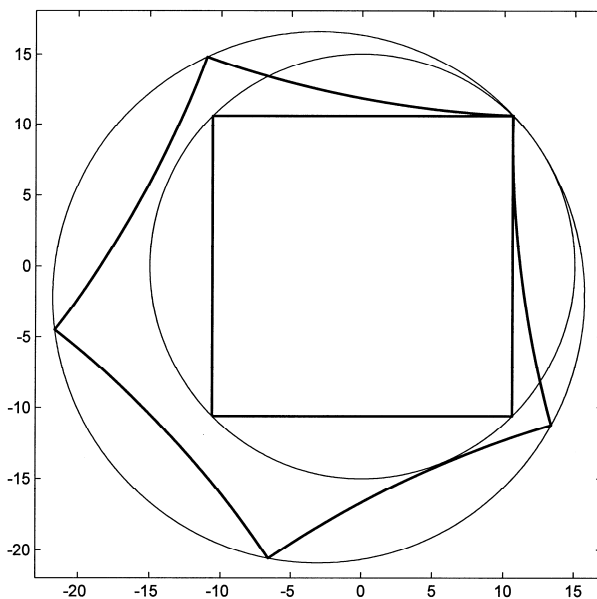


Figura 76. *Reprezentare profiluri (generator și generat în timpul rulării).*

generică din timpul rulării. Exercițiul de rulare se poate repeta pentru

```
l=l+1;
end
plot(xcerc,ycerc,'k',
'LineWidth',
0.5);
axis([-23 17
-22 18])
```

atunci se obține
reprezentarea
din figura 76
(cercurile bază
și rulantă,
profilul
generator și cel
generat, reliefate
și în figura 75)

orice valoare a parametrului N_r (schimbând valoarea variabilei `nlob` din linia 2 a programului PATRAT3). De exemplu, echivalentul figurii 76 pentru $N_r = 7$ se redă în figura 77.

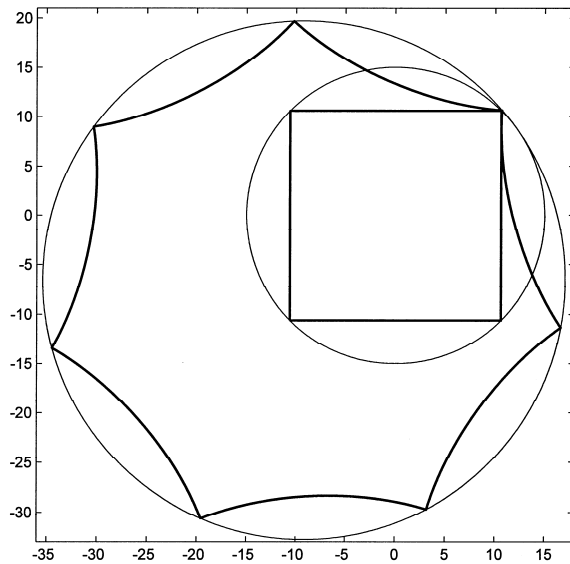


Figura 77. Variantă la figura 76 pentru generarea unui profil pătrat ($N_r = 7$, $N_b = 4$).

Ca și în situațiile anterioare, exercițiul de rulare se poate repeta pentru orice profil plan de generat.

Prezintă interes sinteza sculei pentru generarea arborelui canelat deja abordat în orizontul figurii 23.

Pentru aceasta, prima

parte a programului PATRAT3 (până la instrucțiunea `k=5`; inclusiv) se înlocuiește cu secțiunea de definire a profilului arborelui canelat de definit (reprezentat în figura 19. b.), redată mai jos:

```
clear all;close all;tic
nlob=8;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob/6;x
c=0;yc=0;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-incdef:68*cons;x(k)=9*cos(i);
```

```

y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-
incdef:52*cons;x(k)=10*cos(i);y(k)=10*sin(i);
k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=120*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis equal
k=321;l=1;

```

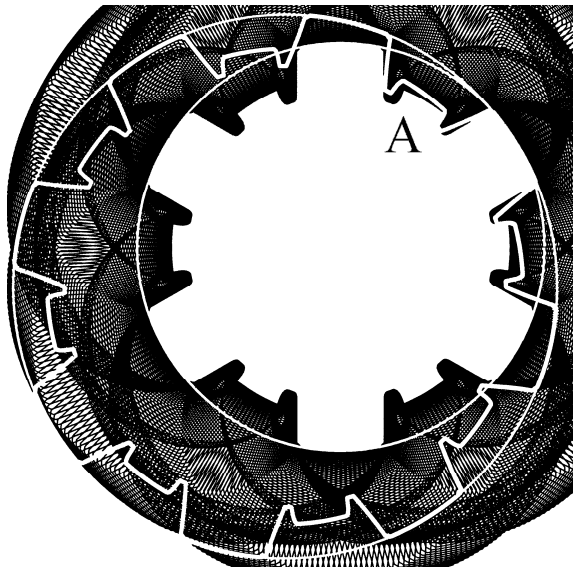


Figura 78. Rezultat generare profil arbore canelat (fig. 19. b.) prin rulare cerc în cerc.

Se obține astfel programul CANELURAM5 (redat complet în Anexa 32) a cărui execuție (cu $N_r = 8$ și $N_b = 6$) conduce conform figurii 78 la obținerea echivalentului grafic al figurii 75.

Un detaliu grafic la figura 78 (zona A) se prezintă în figura 79. Și aici se observă faptul că profilul de arbore este

foarte bine definit pe zonele de cap (arc de cerc și teșire) respectiv de picior (arc de cerc).

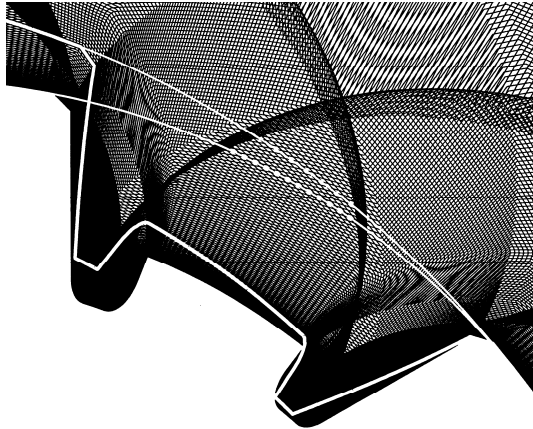


Figura 79. Detaliu grafic la figura 78, în zona A.

canelat din figura 19. a. Pentru aceasta se înlocuiește prima parte a programului PATRAT3 (până la instrucțiunea $k=5$; inclusiv) cu secțiunea:

```
clear all;close all;tic
nlob=8;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob/6;x
c=0;yc=0;
r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=9.6593;x(2)=-2.5882;y(2)=11.5176;
x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;y(15)=11.517
6;
x(16)=2.5882;y(16)=9.6593;k=17;incdef=5.2871*cons/10
;k=17
for i=73.2871*cons:-incdef:68*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=16*cons/16;
```

O comparație cu rezultatul grafic din figura 24 indică clar faptul că degajările pentru eliminarea viciului de rulare (profil de trecere între flanc și picior) sunt aici exagerat de mari.

Este de intuit că viciul de rulare este mai redus în acest caz. Confirmarea acestei constatări se poate obține dacă se simulează generarea prin rulare a profilului de arbore

```

for i=68*cons:-incdef:52*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-incdef:46.7129*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
u=60*cons;for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=120*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=180*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=240*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=300*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end
x(321)=x(1);y(321)=y(1);
plot(x,y);axis equal;k=321;l=1;

```

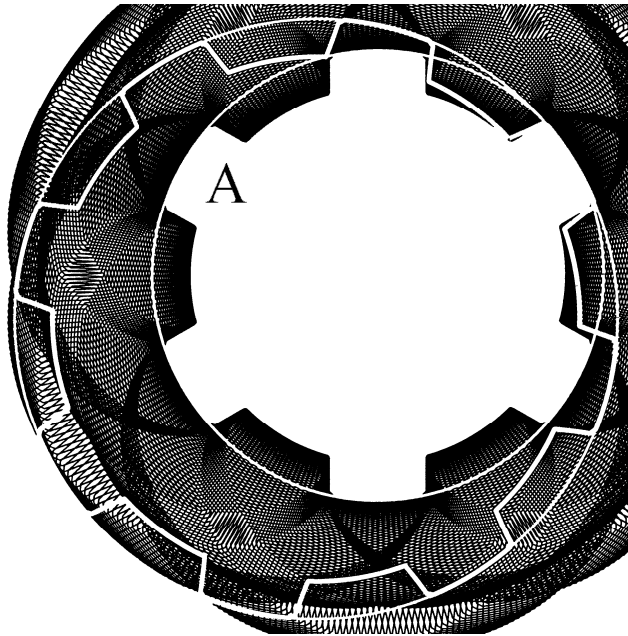


Figura 80. Rezultat generare profil arbore canelat (fig. 19. a.) prin rulare cerc în cerc.

termeni cu cei din figurile 75 și 78). Ca primă observație, se observă că profilul generator este mai simplu. În figura 81 se prezintă un detaliu

Această secțiune descrie profilul de arbore canelat de generat prin rulare ca succesiune de segmente (inclusiv a arcelor de cerc) Compilația se salvează ca program CANELU-RAM6 (redat în Anexa 33).

Rezultatul grafic al execuției programului se prezintă în figura 80 (în aceeași

grafic al figurii 80, din zona A, (fără reprezentarea cercului care joacă rolul bazei).

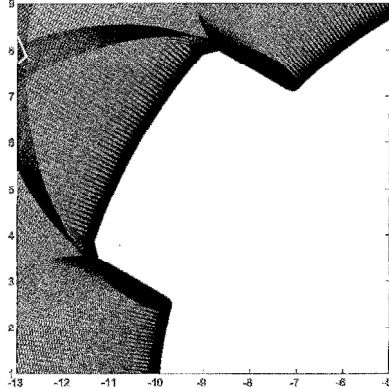


Figura 81. Detaliu grafic la figura 80 în zona A.

O estimare sumară confirmă faptul că la rularea cerc în cerc viciul de rulare (aparitia profilului de trecere la intersecția dintre flanc și arcul de picior) este cu mult mai redus, practic aproape eliminat. Eventualele măsuri de eliminare a viciului de rulare sunt mai puțin severe.

De aici, dincolo de dificultățile reale de transfer în fabricație, rezultă cel puțin acest avantaj al rulării cerc în cerc pentru producerea pieselor de tip arbore. Acest avantaj a fost remarcat și în cazul rulării cerc în cerc pentru producerea pieselor de tip alezaj (la generarea figurii 69).

La fel de bine se poate aplica generarea prin rulare cerc în cerc a profilelor roților dințate danturate exterior folosind un profil generator plasat pe o sculă de tip alezaj (cuțit roată de mortezat cu dantură interioară). Pentru aceasta se depinde programul DETPROF (din Anexa 3) într-un folder separat. Se execută programul (care generează un profil de roată dințată cu 20 de dinți (de altfel redat în figura 6). Se

Pentru comparație se reia în figura 82 un detaliu asemănător al aceleiași caneluri obținute prin rulare, procedeul dreaptă pe cerc (din figura 21).

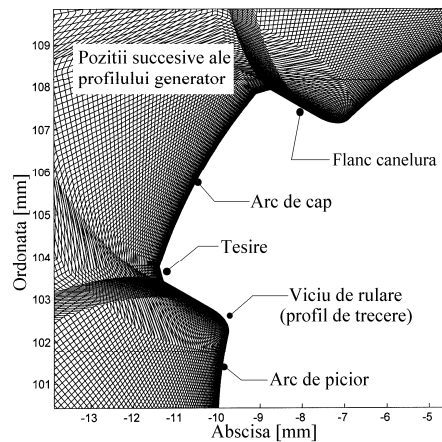


Figura 82. Reluare a figurii 21.

execută apoi instrucțiunile: `save absc;save ord;` Prin aceasta se salvează coordonatele segmentelor care definesc profilul de roată (prin

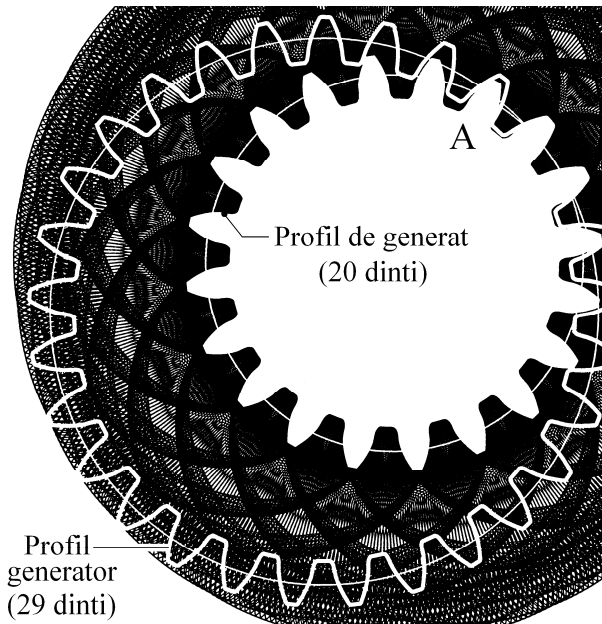


Figura 83. Rezultat generare profil roată dințată danturată exterior prin rulare cerc în cerc (cu sculă de tip alezaj).

aproximare). Aceste coordonate se încarcă de către programul SIMGENDANT3 (redat în Anexa 34) care este structurat de manieră asemănătoare cu programele PATRAT3, CANELURAM5 și CANELURAM6, și care la execuție produce rezultate grafice similare cu cele deja prezentate (deduce profilul generator pentru $N_r = 29$ și $N_b = 20$ și validează acest profil prin

rulare, obținându-se profilul de generat). Rezultatul este expus în figura 83.

Un detaliu al figurii 83 (din zona A) este redat în figura 84. Acesta se obține dacă, după execuția programului SIMGENDANT3 se execută următoarea secvență:

```
close all;incr=2*pi/(100);l=1;
for i=0.2:incr:2.2*pi;
for j=1:k;alfal=alfa+i;
    betal=beta-(rbaz-rrul)/rrul*i;
    x1(j)=(rbaz-
rrul)*cos(alfal)+x(j)*cos(betal)-y(j)*sin(betal);
    y1(j)=(rbaz-
rrul)*sin(alfal)+y(j)*cos(betal)+x(j)*sin(betal);
```

```

        l=l+1;
end
plot(x1,y1,'k','LineWidth',0.5);hold on
end
plot(x1,y1,'k','LineWidth',3.5);
axis equal;axis([-70 170 140 330])

```

Se observă o poziție generică a profilului generator (cu linie groasă de culoare neagră) precum și înfășurătoarea la interior a pozițiilor acestuia (profilul de generat).

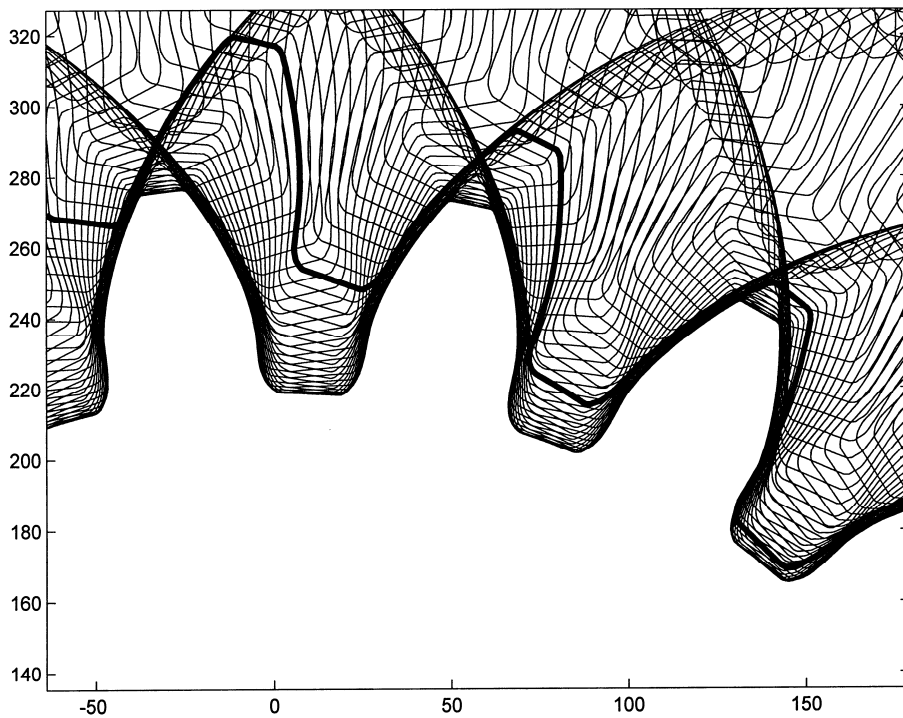


Figura 84. Detaliu grafic la figura 83, în zona A.

Și această imagine este sugestivă pentru înțelegerea aspectelor legate de definirea prin rulare a unei danturi. O posibilă aplicație ar fi legată de studiul pompelor cu roți dințate cu angrenaj interior, la care cele două roți au rol principal de deplasare a fluidului din zona de admisie (joasă presiune) în zona de refulare (întă presiune). Rolul

secundar este acela de a etanșa cele două zone. Calitatea etanșării depinde -printre altele- de calitatea contactului profilelor în zona de angrenare (cu joc minim).

Dacă acum se rulează secvența următoare:

```
close all
plot(x1,y1,'k','LineWidth',1.5);
axis equal;hold on;
plot(xroata,yroata,'k','LineWidth',1.5);
```

atunci se obține reprezentarea din figura 85 care descrie poziția relativă a profilelor generator și de generat pe parcursul rulării.

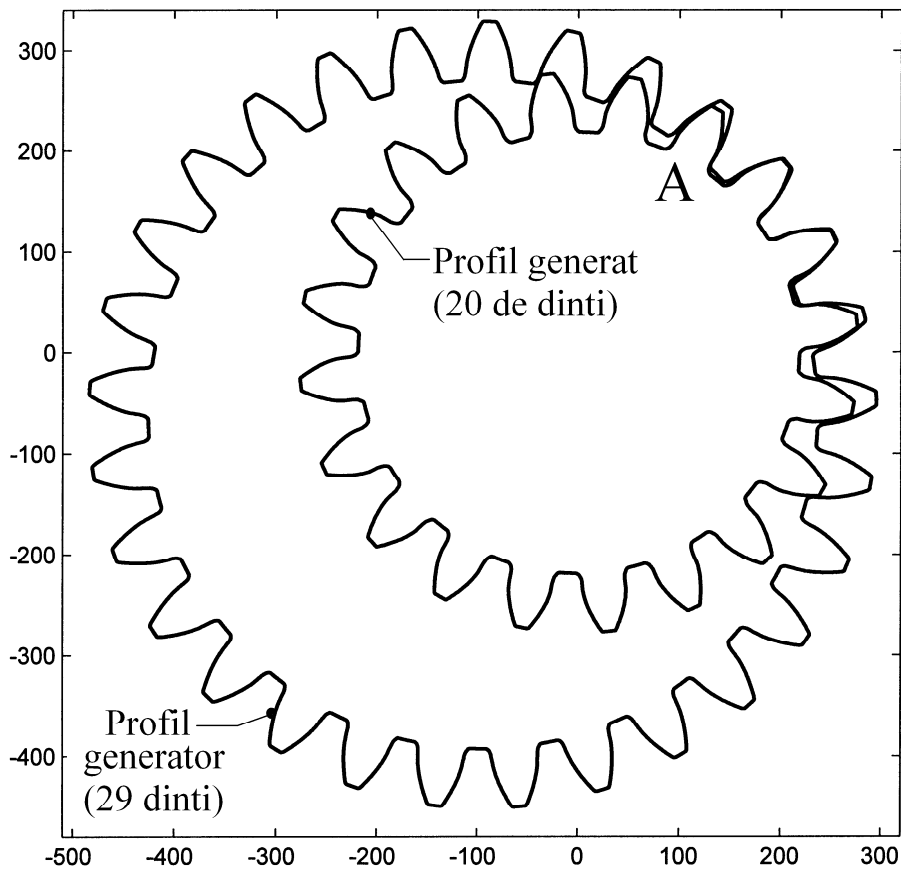


Figura 85. Reprezentare a profilului generat și generator (din figura 83).

Aparent s-a obținut același gen de reprezentare ca în figura 72. Excepția aici este legată de faptul că profilul generator se află la exterior iar cel generat se află la interior.

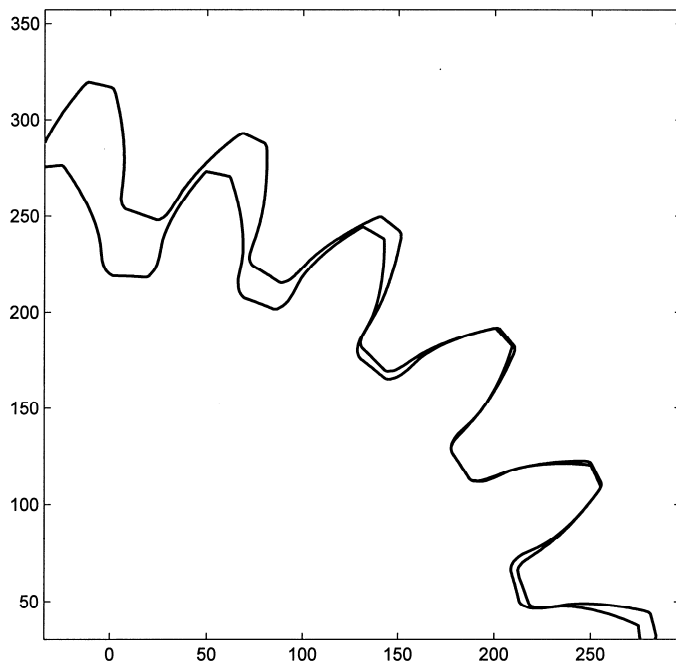


Figura 86. Detaliu grafic la reprezentarea din figura 85 (zona A).

În figura 86 se prezintă un detaliu la figura 85, în zona A, de „contact-an-grenare” a celor două profile. Și această reprezentare poate servi foarte bine înțelegerii naturii contactului dintre profilul generator și cel generat.

Una dintre finalitățile acestui demers ar fi cercetarea apariției fenomenului de subțaiere la generarea roților dințate cu număr mic de dinți prin rularea cerc în cerc.

Fie necesitatea generării prin rulare cerc în cerc (cu sculă de tip alezaj) a unui profil de roată dințată cu patru dinți, obținut grafic prin rularea programului redat în continuare (și în Anexa 35). Pentru simplificare s-a considerat că flancurile dinților acestui profil sunt arce de cerc.

PROGRAM FIG87

```
close all;clear all;
r=27.5;a=100;b=100;nincr=30;k=1;
%GENERARE DINTE 1 %Generare arc de flanc stanga;
nincr=30;k=1;
```

```

for i=0:nincr;constx1=21.5315;consty1=6.6331;
    limunghi1=158.7645*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de cap;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=75*pi/180;limunghi2=limunghi1-
(15*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+25*cos(limunghi1+i*incrunghi);
    y(k)=consty1+25*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de flanc dreapta;
for i=0:nincr;constx1=-10.5348;consty1=19.9154;
    limunghi1=4.308*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de picior;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=45*pi/180;limunghi2=limunghi1-
(45*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+15*cos(limunghi1+i*incrunghi);
    y(k)=consty1+15*sin(limunghi1+i*incrunghi);
k=k+1;
end
k=k-1;
plot(x,y);hold on;axis equal
l=1;

```

```

for j=0:3;
alfa=cos(-j*pi/2);beta=sin(-j*pi/2);
for i=1:k;
    x1(l)=x(i)*alfa-y(i)*beta;
    y1(l)=y(i)*alfa+x(i)*beta;l=l+1
end
end
l=l-1;clear x;clear y;x=x1;y=y1;k=1;l=1;
plot(x,y,'k','LineWidth',1.5);axis([-30 30 -26 26])

```

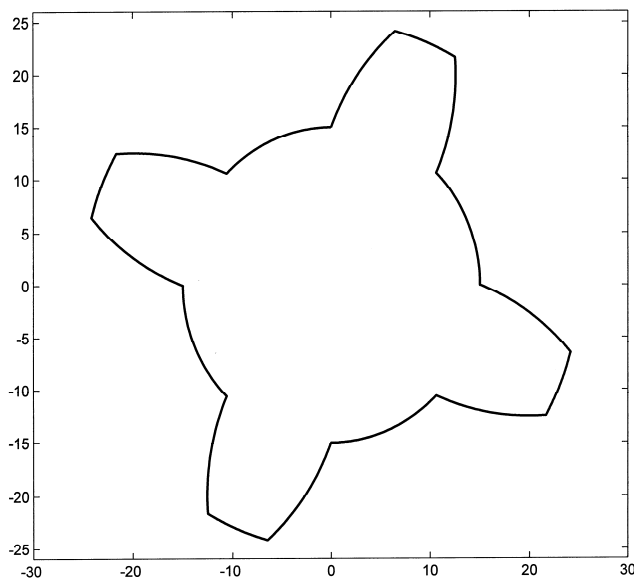


Figura 87. *Reprezentare a unui profil transversal (convențional) de roată dințată cu patru dinți.*

Rularea programului produce rezultatul grafic din figura 87, cu descrierea profilului transversal al roții de generat. S-a considerat că flancurile și zonele de fund ale dințelului (ambele arce de cerc) nu sunt

racordate. Fiecare dintre cele 16 arce de cerc este aproximat printr-o succesiune de 30 de segmente (coarde de arc de cerc).

Se pune problema generării unui astfel de profil prin rulare cerc în cerc cu sculă de tip alezaj. Pe baza considerentelor teoretice deja folosite anterior se realizează deducerea prin rulare a profilului generator, ($N_b = 4$, $N_r = 6$), urmată de rularea acestuia în scopul obținerii profilului de generat. Se realizează sinteza programului SIMGENDANT4, cu listingul complet redat în ANEXA 36.

Rularea programului conduce la rezultatul grafic reprodus în figura 88.

O pozitie a
profilului
generator

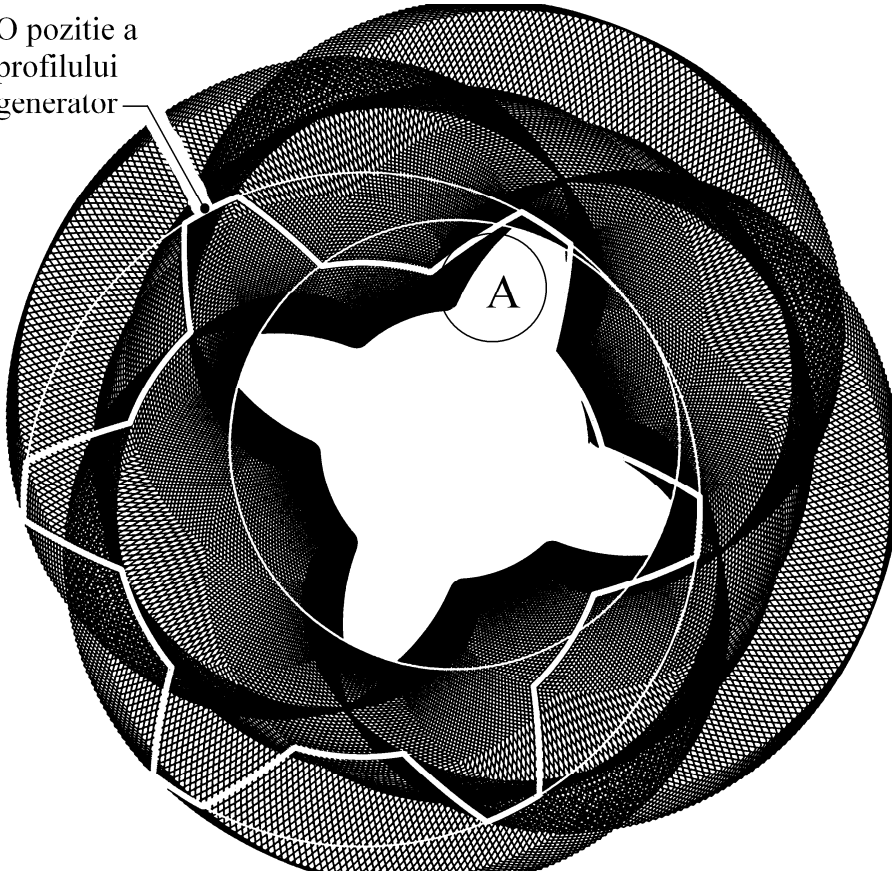


Figura 88. Rezultat generare profil roată dințată danturată exterior (cu număr mic de dinți) prin rulare cerc în cerc (cu sculă de tip alezaj).

Cel puțin la o primă evaluare, se observă faptul că, aparent, profilul de generat (ca înfășurată la interior a pozițiilor succesive ale profilului generator) rezultă fără subțiere la baza dintelui. Aceasta, chiar dacă rularea se face cu cercuri de bază și de rulare tangente la cercurile de cap ale profilelor.

Programul SIMGENDANT4 are o particularitate legată de timpul de execuție: cca. 20 minute. Durata mare de execuție este condiționată de aproximarea fiecăruia dintre cele 16 arce de cerc ale

profilului de generat prin cate o succesiune de 30 de segmente (coarde pe arc).

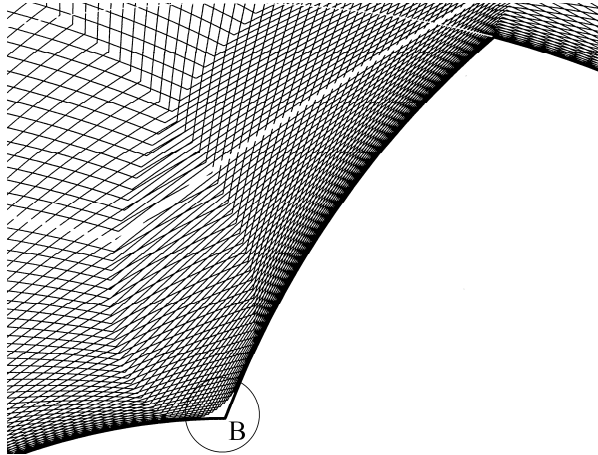


Figura 89. Detaliu al figurii 88 (în zona A).

de generat (teoretic și real) practic coincid. Excepția este minoră,

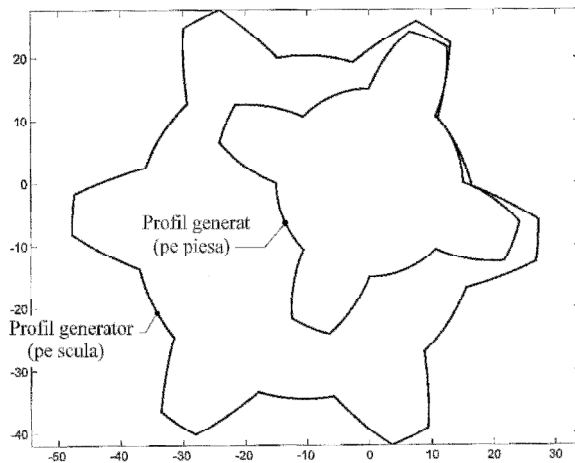


Figura 90. O poziție relativă a celor două profile pe parcursul rulării.

ea este marcată în zona B, de racordare dintre flancul și fundul (piciorul) dintelui. Este acesta un rezultat important care poate fi luat în calcul pentru eliminarea neajunsului clasic de la rularea cerc pe dreaptă sau cerc pe cerc: aparitia subtăierii la baza dintelui. Evident, acest avantaj trebuie evaluat în contrapondere cu dificultățile de realizare a sculei de tip alezaj și, evident, cu dificultățile de implementare a procedurii de rulare.

Un detaliu al figurii 88 în zona A este redat în figura 89. Peste profilul de generat real (rezultat ca înfășurătoare a pozițiilor succesive ale profilului generator) se suprapune profilul de generat teoretic.

Apare evident faptul că cele două profile

de generat (teoretic și real) practic coincid. Excepția este minoră, ea este marcată în zona B, de racordare dintre flancul și fundul (piciorul) dintelui. Este acesta un rezultat important care poate fi luat în calcul pentru eliminarea neajunsului clasic de la rularea cerc pe dreaptă sau cerc pe cerc: aparitia

Dacă ulterior execuției programului SIMGENDANT4 se introduce în fereastra de comandă Matlab și se execută secvența:

```
close all  
plot(x1,y1,'k','LineWidth',1.5);axis equal;hold on;  
plot(xroata,yroata,'k','LineWidth',1.5);
```

atunci se obține reprezentarea grafică a unei poziții a profilului generator și a profilului de generat (teoretic) pe parcursul rulării, conform figurii 90, aceeași cu cea deja evidențiată în figura 88.

9. Concluzii

Înțelegerea, validarea și confirmarea procedurilor tehnice și tehnologice [16, 19] care concură la generarea prin rulare a curbelor plane este favorizată pe deplin de facilitățile de simulare asistată de calculator prin aplicații concrete. Toate considerațiile teoretice legate de rulare, transpuse în relații analitice (ecuații parametrice) pot fi pe deplin instrumentate și utilizate în realitatea virtuală folosind mediile de programare adecvate (aici Matlab). Sunt valorizate avantajele simulării numerice ca auxiliar al cunoașterii și ca sistem de verificare-validare *ad-hoc* al tuturor demersurilor teoretice legate de acest subiect (în principal emularea tăișului de sculă și transpunerea acestuia pe piesă, ambele prin rulare). Este ușor de imaginat că în absența simulării numerice validarea premiselor teoretice s-ar face, cu costuri și riscuri semnificativ mai mari, direct pe mașina-unealtă, pe proces real [14, 18].

Cititorul poate dezvolta problematica expusă aici, găsind noi direcții de exercițiu. Evident, simularea curbelor plane (2D) folosite ca generatoare pe suprafețe cu directoare dreaptă sau elicoidală oferă cheia înțelegerii definirii -în principal- a suprafețelor de angrenare pentru roțile dințate cilindrice (cu dinți simetrici sau asimetrici [15]). Folosind acest punct de plecare cititorul interesat va putea aborda problema

generării suprafețelor de angrenare definite cu generatoare (directoare) 3D [3] (regăsite de exemplu pe roțile dințate conice cu dantură în arc de cerc). Aceasta va fi -de altfel- una dintre direcțiile viitoare de cercetare pentru autorul prezentei lucrări.

Bibliografie

1. Biran, A., Breiner, M., (1999), *Matlab 5 for Engineers* (2nd Edition), Prentice Hall.
2. Constantin, V., Palade, V., (2005), *Organe de mașini și mecanisme, Vol. II, Transmisii mecanice*, Editura Fundației Universitare „Dunărea de Jos” din Galați.
3. Dimitriou, V., Antoniadis, A., (2009), *CAD-based simulation of the hobbing process for the manufacturing of spur and helical gears*, International Journal of Advanced Manufacturing Technology, vol.41, pp. 347-357.
4. Gafițanu, M. et. al. (1983), *Organe de mașini vol. II*, Editura Tehnică, București.
5. Gavrițaș, I., Voicu, N., (1982), *Tehnologia de fabricație a roților dințate pe mașini-unelte clasice și cu comandă program*, Editura Tehnică, București.
6. Gheghea, I., Plahteanu, B., Mitoșeriu, C., Ghionea, A., *Mașini-unelte și agregate*, (1983), Editura Didactică și Pedagogică, București.
7. Ghinea, M., Firețeanu, V., (2003) *MATLAB-calcul numeric, grafică, aplicații*, Teora, București.
8. Grămescu, T., Slătineanu, L., Pruteanu, O., (1993), *Tehnologii de danturare a roților dințate*, Editura Universitas, Chișinău.
9. Horodincă, M., (1998) *Elements of Geometrical Modelling for Tooling Allowance (Cutter Chip Surface) Forming, Setting and Describing Implication of Tool In Cylindrical Toothed Wheel Processing*, Buletinul I. P. Iași (publicat de U.T. Iași), Secțiunea Construcții de Mașini, Tomul XLIV(XLVIII), fasc. 1-2, p. 103-120.
10. Horodincă, M., (1998), *Contribuții la optimizarea sistemelor de comandă adaptivă și diagnoză la mașinile de frezat roți dințate*, Teză de doctorat, Universitatea Tehnică Gh. Asachi din Iași.

11. Horodincă, M., (2013) *Monitorizare, diagnoză și conducere adaptivă pe mașini de frezat roți dințate*, Editura Performantica, Iași.
12. Maroș, D., (1958), *Teoria mecanismelor și a mașinilor. Cinematica roților dințate*, Editura Tehnică, București.
13. Minciu, C., (1986), *Proiectarea și tehnologia sculelor pentru danturare*, Editura Tehnică, București.
14. Moraru, V., Plahteanu B., Velicu S., Aurite Tr. (1982), *Mașini-unelte speciale*, Editura Didactică și Pedagogică, București.
15. Ravai Nagy, S., (2012), *Contribuții la dimensionarea, testarea și execuția roților dințate cu dinți asimetrici*, Teză de doctorat, Univesritatea Tehnică din Cluj-Napoca, Centrul universitar de Nord din Baia Mare.
16. Sauer, L., (1986), *Tehnologia de fabricație a roților dințate*, București, Editura Tehnică.
17. Stoica, I. A., (1977) *Interferența roților dințate*, Editura Dacia, Cluj-Napoca.
18. Vaida, A., Botez, E., Velicu, S. (1980), *Proiectarea mașinilor-unelte*, Editura Didactică și Pedagogică, București.
19. Vlase, A., Opran, C., Cristea, R., Georghiu S., (1998), *Tehnologii de prelucrare pe mașini de danturat*, Editura Tehnică, București.

Anexa 1**PROGRAM TRIUNGHI**

```

close all;clear all
r=10;a=100;b=0;
x(1)=0;y(1)=10;
x(2)=8.660254;y(2)=-5;
x(3)=-8.660254;y(3)=-5;
x(4)=0;y(4)=10;
k=4;l=1;
for i=0:2*pi/50:2*pi;u=i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)-y(j)*sin(u);
    y1(l)=b+x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
plot(x1,y1,'k');hold on
axis equal;
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end;k=k-1;
inc2=(maxx-minx)/3000;l=1;minminy=10000;
for i=minx+0.001:inc2:maxx-inc2;miny=1000;ics=0;
    abs(l)=i;

```

```
    for j=1:k-1;
        if i>x1(j);if i<x1(j+1);int=m(j)*i+n(j);
            int;if int<miny;ord(l)=int;
miny=int;
            else end;else end;else end
        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
            int;if int<miny;ord(l)=int;
miny=int;
            else end;else end;else end
    end;l=l+1;
    if miny<minminy;minminy=miny;else end
end
dim1=size(ord);dim=dim1(2)
plot(abs,ord,'k','LineWidth',1.5)
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',2)
```

Anexa 2**PROGRAM SIMGENDANT**

```

clear all;close all;
m=25;z=20;a=120;fi=90*pi/180;
alfa=20*pi/180;c=2.35*m*tan(alfa);
b=0.5*m*(pi-4.7*tan(alfa));dep=0*m;
rr=0.5*m*(z);x(1)=a;x(2)=a+b;
x(3)=a+b+c;x(4)=x(3)+b;
x(5)=x(4)+c;x(6)=x(5)+b;
y(1)=1.1*m+dep;y(2)=1.1*m+dep;
y(3)=-1.25*m+dep;y(4)=-1.25*m+dep;
y(5)=1.1*m+dep;y(6)=1.1*m+dep;
ndinti=z+5;for j=1:ndinti;
for i=1:6;x(j*6+i)=x(i)+j*m*pi;
y(j*6+i)=y(i);end;end; xc=0;yc=0;
liminfteta=0;limsupteta=2.5*pi;inc=2*pi/100;
for teta=liminfteta:inc:limsupteta;
for i=1:6*(ndinti)
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
end
plot(x1,y1,'k');hold on;axis equal
end
axis([- (rr+1.2*m), (rr+1.2*m), -
(rr+1.2*m), (rr+1.2*m)])
Title('REZULTATUL GRAFIC AL SIMULARII GENERARII
DANTURII PRIN RULARE')
xlabel('Excursia pe abscisa sistemului de
reprezentare 2D [mm]')
ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')

```

Anexa 3**PROGRAM DETPROF**

```

clear all;close
all;m=25;z=20;a=120;fi=90*pi/180;m1=m;
alfa=20*pi/180;c=2.35*m*tan(alfa);
b=0.5*m*(pi-4.7*tan(alfa));dep=0*m;
rr=0.5*m*(z);
x(1)=a;x(2)=a+b;x(3)=a+b+c;x(4)=x(3)+b;
x(5)=x(4)+c;x(6)=x(5)+b;
y(1)=1.1*m+dep;y(2)=1.1*m+dep;
y(3)=-1.25*m+dep;y(4)=-1.25*m+dep;
y(5)=1.1*m+dep;y(6)=1.1*m+dep;ndinti=z+5;
for j=1:ndinti;
    for i=1:6;x(j*6+i)=x(i)+j*m*pi;
y(j*6+i)=y(i);
end;end
xc=0;yc=0;
l=1;
liminf teta=0;limsup teta=2.5*pi;
inc=2*pi/300+0.00001;
for teta=liminf teta:inc:limsup teta;
for i=1:6*(ndinti)
x1(l)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(l)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);l=l+1;
end
end;l=l-1;
%plot(x1,y1,'k');hold on;
Title('REZULTATUL GRAFIC AL SIMULARII GENERARII
DANTURII PRIN RULARE')
xlabel('Excursia pe abscisa sistemului de
reprezentare 2D[mm]')
ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')
%rutina de determinare a profilului
k=1;

```



```

for i=1:l-1;
dif=(x1(i+1)-x1(i));if
dif==0;dif=0.00000000001;else
end
m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);k=k+1;
end; k=k-1;
inc=2*pi/2000;l=1;
for i=0:inc:2.1*pi;mindist=1000;
xa=xc;ya=yc;xb=xa+5*cos(i);yb=ya+5*sin(i);
dif=xb-xa;
if dif==0;dif=0.00000000001;else end;
mi=(yb-ya)/dif;ni=ya-xa*mi;
for j=1:k-1;dif1=mi-m(j);
    if dif1==0;dif1=0.00000000001;else end;
intx=(n(j)-ni)/dif1;inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
if dist1<dist;
if intx==x1(j+1);if inty==y1(j+1);
    if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else
end;else end;else end;
if intx==x1(j);if inty==y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else
end;else
end;else end;
if intx<x1(j+1);if intx>x1(j);if
dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else
end;else end;else end;
if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
if inty>y1(j+1);if inty<y1(j);

```

```
if dist<mindist;absc(1)=intx;ord(1)=inty;  
mindist=dist;else end;else end;else end;  
else end  
end;  
l=l+1;  
end  
plot(absc,ord,'k','LineWidth',1.5);  
axis equal  
axis([- (rr+1.4*m1), (rr+1.4*m1), -  
(rr+1.4*m1), (rr+1.4*m1)])
```

Anexa 4**PROGRAM SIMGENDANTCURBEL**

```

clear all;close all;
m=25;z=20;a=120;fi=90*pi/180;
alfa=20*pi/180;c=2.25*m*tan(alfa);b=0.5*m*(p
i-4.5*tan(alfa));dep=0*m;rr=0.5*m*(z);
xa(1)=a;xa(2)=a+b;xa(3)=a+b+c;xa(4)=xa(3)+b;
xa(5)=xa(4)+c;xa(6)=xa(5)+b;
ya(1)=1.1*m+dep;ya(2)=1.1*m+dep;ya(3)=-
1.25*m+dep;ya(4)=-
1.25*m+dep;ya(5)=1.1*m+dep;ya(6)=1.1*m+dep;
g=1;
x1=xa(1);y1=ya(1);x2=xa(2);y2=ya(2);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;i=1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;
end
x1=xa(2);y1=ya(2);x2=xa(3);y2=ya(3);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;
end
x1=xa(3);y1=ya(3);x2=xa(4);y2=ya(4);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;
end
x1=xa(4);y1=ya(4);x2=xa(5);y2=ya(5);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;
for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;
end
x1=xa(5);y1=ya(5);x2=xa(6);y2=ya(6);
m1=(y2-y1)/(x2-x1);n=y1-m1*x1;

```

```

for a=x1:g:x2;x(i)=a;y(i)=m1*x(i)+n;i=i+1;
end
npuncte=i-1;
for j=1:z+5;
    for i=1:npuncte;
x(j*npuncte+i)=x(i)+j*m*pi;
y(j*npuncte+i)=y(i);
    end
    end
u1=size(x);u=u1(2);
xc=0;yc=0;
liminf_teta=0;limsup_teta=1000*pi/180;
inc=360/(10*z)*pi/180;k=1;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:u
        x1a(k,i)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
        y1a(k,i)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);
    end;k=k+1;
end
plot(x1a,y1a,'k');axis equal
axis([- (rr+1.6*m), (rr+1.6*m), -
(rr+1.6*m), (rr+1.6*m)])
xlabel('Abscisa reprezentarii [mm]');
ylabel('Ordonata reprezentarii [mm]')

```

Anexa 5**PROGRAM TRIUNGHI1**

```

close all;clear all
r=10;a=100;b=0;x(1)=0;y(1)=10;x(2)=8.660254;
y(2)=-5;x(3)=-8.660254;y(3)=-5;
x(4)=0;y(4)=10;k=4;l=1;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.000000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=10:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;

```

```

else end;else end;else end
    if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
%plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+5;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminfteta=0;limsupteta=3*pi;inc=2*pi/100;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])

```

Anexa 6**PROGRAM PATRAT1**

```

close all;clear all;r=10*2^0.5;a=100;b=100;
x(1)=-10;y(1)=-10;x(2)=10;y(2)=-
10;x(3)=10;y(3)=10;
x(4)=-10;y(4)=10;x(5)=-10;y(5)=-10;k=5;l=1;

for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=-60:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;

```

```

        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
        if int<miny;ord(l)=int;
miny=int;
        else end;else end;else end
        if i<x1(j);if
i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+5;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminf_teta=0;limsup_teta=3*pi;inc=2*pi/300;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])

```


Anexa 7**PROGRAM PENTAGON1**

```

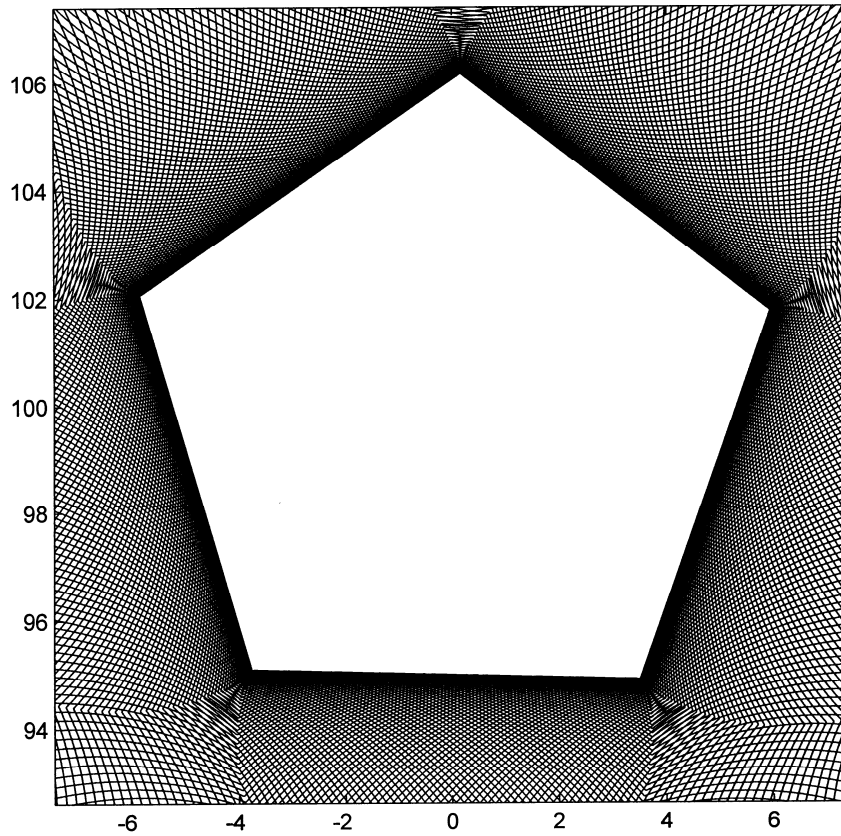
close all;clear all;r=6.18;a=100;b=100;
x(1)=-5.8779;y(1)=1.9098;x(2)=-3.6327;
y(2)=-5;x(3)=3.6327;y(3)=-5;
x(4)=5.8779;y(4)=1.9098;x(5)=0;y(5)=6.1803;
x(6)=-5.8779;y(6)=1.9098;k=6;l=1;
for i=0:3*pi/300:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end;
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end;
k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=45:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;

```

```

        if int<miny;ord(l)=int;
miny=int;
        else end;else end;else end
        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;if miny<minminy;minminy=miny;
else end
end

```



Rezultatul grafic al execuției programului PENTAGON1.

```

dim1=size(ord);dim=dim1(2);
%plot(abs,ord,'r','LineWidth',1.5);axis equal
ord=minminy-ord;

```

```
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+5;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminfteta=0;limsupteta=5*pi;inc=2*pi/300;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])
```

Anexa 8**PROGRAM HEXAGON1**

```

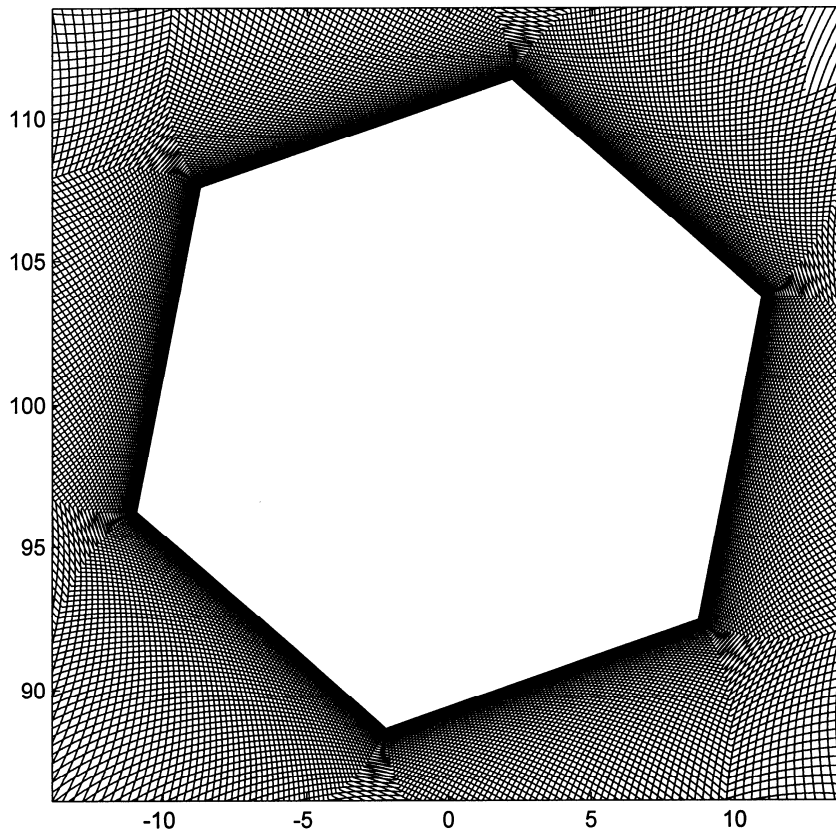
close all;clear all;r=11.5470;a=100;b=100;
x(1)=-5.7735;y(1)=10;x(2)=-11.5470;y(2)=0;x(3)=-
5.7735;y(3)=-10;
x(4)=5.7735;y(4)=-
10;x(5)=11.5470;y(5)=0;x(6)=5.7735;y(6)=10;
x(7)=-5.7735;y(7)=10;k=7;l=1;
for i=0:3*pi/300:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=0:inc2:80;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;

```

```

                                if int<miny;ord(l)=int;
miny=int;
                                else end;else end;else end
                                if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
                                if int<miny;ord(l)=int;
miny=int;
                                else end;else end;else end
                                end;l=l+1;
                                if miny<minminy;minminy=miny;
                                else end
                                end

```



Rezultatul grafic al execuției programului HEXAGON1.

```
dim1=size(ord);dim=dim1(2);
```

```
%plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+5;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminfteta=0;limsupteta=5*pi;inc=2*pi/300;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])
```

Anexa 9**PROGRAM fig16**

```

close all;clear all
r=10;a=100;b=0;a1=0;a2=2.8868;a3=8.6603;
a4=5.7735;b1=10;b2=5;b3=0;
x(1)=a1;x(2)=a2;x(3)=a3;x(4)=a4;x(5)=a3;
x(6)=a2;x(7)=a1;x(8)=-a2;
x(9)=-a3;x(10)=-a4;x(11)=-a3;
x(12)=-a2;x(13)=a1;
y(1)=b1;y(2)=b2;y(3)=b2;
y(4)=b3;y(5)=-b2;y(6)=-b2;y(7)=-b1;y(8)=-b2;
y(9)=-b2;y(10)=b3;y(11)=b2;y(12)=b2;y(13)=b1;
k=13;l=1;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;

```

```

for i=10:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;
                else end;else end;else end
                    if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
%plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+6;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminfteta=0;limsupteta=3*pi;inc=2*pi/200;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
        end
        plot(x1,y1,'k');hold on;axis equal
    end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])

```


Anexa 10**PROGRAM fig18**

```

close all;clear all
r=10;a=100;b=0;a1=0;a2=2.8868;a3=8.6603;
a4=5.7735;b1=10;b2=5;b3=0;
x(1)=a1;x(2)=a2;x(3)=a3;x(4)=a4;x(5)=a3;
x(6)=a2;x(7)=a1;x(8)=-a2;
x(9)=-a3;x(10)=-a4;x(11)=-a3;
x(12)=-a2;x(13)=a1;
y(1)=b1;y(2)=b2;y(3)=b2;
y(4)=b3;y(5)=-b2;y(6)=-b2;y(7)=-b1;y(8)=-b2;
y(9)=-b2;y(10)=b3;y(11)=b2;y(12)=b2;y(13)=b1;
k=13;l=1;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
%plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;

```

```

prag=-6.1;
for i=10:inc2:90;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;if ord(l)>prag;ord(l)=-5.8;else end
                else end;else end;else end
                    if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
            if int<miny;ord(l)=int;
miny=int;if ord(l)>prag;ord(l)=-5.8;else end
else end;else end;else end

end;
l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+6;y=ord;rr=r;fi=pi/2;xc=0;yc=100;
liminfteta=0;limsupteta=3*pi;inc=2*pi/400;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])

```

Anexa 11**PROGRAM CANELURAM1**

```

close all;clear
all;r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=9.6593;x(2)=-
2.5882;y(2)=11.5176;
x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;
x(16)=2.5882;y(16)=9.6593;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-incdef:68*cons;
x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-incdef:52*cons;
x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
u=60*cons;for i=1:53;
x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);
k=k+1;end;u=120*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);
k=k+1;end;u=180*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);

```

```

y(k)=-x(i)*sin(u)+y(i)*cos(u);
k=k+1;end;u=240*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);
k=k+1;end;u=300*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end
x(321)=x(1);y(321)=y(1);
plot(x,y);axis equal;k=321;l=1;
  for i=0:2*pi/500:5/2*pi;u=-i;for j=1:k;
x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
y1(l)=b-x(j)*sin(u)+y(j)*cos(u);
l=l+1;end;end;l=l-1;
plot(x1,y1,'b');hold on;
axis equal;maxx=-100;minx=100;
for i=1:l;if x1(i)>maxx;maxx=x1(i);else end
if x1(i)<minx;minx=x1(i);else end
x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;
  k=1;
for i=1:l-1;
  dif=(x1(i+1)-x1(i));
  if dif==0;dif=0.000000000001;else end
  m(k)=(y1(i+1)-y1(i))/dif;
  n(k)=y1(i)-m(k)*x1(i);
  k=k+1;
end; k=k-1;k
inc2=(maxx-minx)/4000;l=1;minminy=10000;
%for i=minx+0.001:
inc2:maxx-inc2;miny=1000;ics=0;
  for i=10:inc2:90;miny=1000;abs(l)=i;
  for j=1:k-1;
    if i>x1(j);if
i<x1(j+1);int=m(j)*i+n(j);int;
if int<miny;ord(l)=int;miny=int;
else end;else end;else end

```

```

        if i<x1(j);if
i>x1(j+1);int=m(j)*i+n(j);int;
if int<miny;ord(l)=int;miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);axis equal
ord=minminy-ord;
close all;plot(abs,ord,'k','LineWidth',1.5);
axis equal;
hold on
clear x1;clear y1;clear x;clear
y;x=abs+13;y=ord;
fi=pi/2;
xc=0;yc=100;
liminf_teta=0;limsup_teta=3*pi;inc=2*pi/500;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:dim
        x1(i)=xc+r*cos(fi-teta)+(x(i)-
r*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
        y1(i)=yc+r*sin(fi-teta)+(x(i)-
r*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);
    end
    plot(x1,y1,'k');hold on;
axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])

```

Anexa 12**PROGRAM CANELURAM2**

```

close all;clear
all;tic;r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17
;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-
incdef:68*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-
incdef:52*cons;x(k)=10*cos(i);y(k)=10*sin(i)
;
k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=120*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;

```

```

    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis
equal
k=321;l=1;
for i=0:2*pi/500:5/2*pi;u=-i;for j=1:k;
x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
y1(l)=b-
x(j)*sin(u)+y(j)*cos(u);l=l+1;end;end;l=l-1;
plot(x1,y1,'b');hold on;axis equal;maxx=-
100;minx=100;
for i=1:l;if x1(i)>maxx;maxx=x1(i);else end
if x1(i)<minx;minx=x1(i);else end
x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;
    k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.000000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc2=(maxx-minx)/4000;l=1;minminy=10000;
%for i=minx+0.001:inc2:maxx-
inc2;miny=1000;ics=0;
    for i=10:inc2:90;miny=1000;abs(l)=i;
    for j=1:k-1;
        if i>x1(j);if
i<x1(j+1);int=m(j)*i+n(j);int;if

```

```

int<miny;ord(l)=int;miny=int;else end;else
end;else end
    if i<x1(j);if
i>x1(j+1);int=m(j)*i+n(j);int;if
int<miny;ord(l)=int;miny=int;else end;else
end;else end
end;l=l+1;
if miny<minminy;minminy=miny;else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);axis equal
ord=minminy-ord;
close
all;plot(abs,ord,'k','LineWidth',1.5);axis
equal;
hold on
clear x1;clear y1;clear x;clear
y;x=abs+13;y=ord;
fi=pi/2;
xc=0;yc=100;
liminf_teta=0;limsup_teta=3*pi;inc=2*pi/500;
for teta=liminf_teta:inc:limsup_teta;
    for i=1:dim
        x1(i)=xc+r*cos(fi-teta)+(x(i)-
r*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
        y1(i)=yc+r*sin(fi-teta)+(x(i)-
r*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])
toc

```


Anexa 13**PROGRAM DETPROFA**

```

clear all;close all;
m=25;z=20;a=120;fi=90*pi/180;m1=m;
alfa=20*pi/180;c=2.35*m*tan(alfa);
b=0.5*m*(pi-4.7*tan(alfa));dep=0*m;rr=0.5*m*(z);
x(1)=a;x(2)=a+b;x(3)=a+b+c;x(4)=x(3)+b;
x(5)=x(4)+c;x(6)=x(5)+b;
y(1)=1.1*m+dep;y(2)=1.1*m+dep;y(3)=-1.25*m+dep;
y(4)=-1.25*m+dep;y(5)=1.1*m+dep;
y(6)=1.1*m+dep;ndinti=z+5;
for j=1:ndinti;
for i=1:6;x(j*6+i)=x(i)+j*m*pi;
y(j*6+i)=y(i);end;end
xc=0;yc=0;l=1;liminf teta=0;
limsup teta=2.5*pi;inc=2*pi/300+0.00001;
for teta=liminf teta:inc:limsup teta;
    for i=1:6*(ndinti)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);l=l+1;
    end
end;l=l-1;%plot(x1,y1,'k');hold on;
axis equal
axis([- (rr+1.4*m1), (rr+1.4*m1), -
(rr+1.4*m1), (rr+1.4*m1)])
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);

```

```

        k=k+1;
end; k=k-1;
inc=2*pi/2000;l=1;
for i=0:inc:2.1*pi;mindist=1000;
xa=xc;ya=xc;xb=xa+5*cos(i);yb=ya+5*sin(i);
dif=xb-xa;if dif==0;dif=0.00000000001;
else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
    intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
    dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
    if dist1<dist;
        if intx==x1(j+1);
if inty==y1(j+1);if
dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if intx==x1(j);if inty==y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);

```

```

clear x;clear y;x=absc;y=ord;clear absc;clear
ord;clear x1;clear y1;
dim1=size(x);k=dim1(2);l=1;r=rr
for i=0:2*pi/100:2*pi;u=i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)-y(j)*sin(u);
y1(l)=b+x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end;l=l-1;
plot(x1,y1,'k');hold on;axis equal;
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);k=k+1;
end; k=k-1;
inc2=(maxx-minx)/3000;l=1;minminy=10000;
for i=minx+0.001:inc2:
maxx-inc2;miny=1000;ics=0;abs(l)=i;
    for j=1:k-1;
        if i>x1(j);if
i<x1(j+1);int=m(j)*i+n(j);int;
if int<miny;ord(l)=int;miny=int;
else end;else end;else end
        if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
if int<miny;ord(l)=int;miny=int;else end;else
end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'k','LineWidth',1.5);axis equal

```

Anexa 14**PROGRAM SIMURLAN**

```

clear all;close all;m=10;z=10;
fi=90*pi/180;m1=m;rr=0.5*m*(z);dep=0*m;
x(1)=-5.1816;y(1)=20.7264+dep;x(2)=-5.1816;
y(2)=8.5528+dep;k=3;cons=pi/180;max=58.79102;
min=-238.79102;inc=(max-min)/30;
for i=min:inc:max;x(k)=10*cos(i*cons);
y(k)=10*sin(i*cons)+dep;k=k+1;end;
x(k)=5.1816;y(k)=8.5528+dep;x(k+1)=5.1816;
y(k+1)=20.7266+dep;x(k+2)=26.2343;
y(k+2)=20.7264+dep;k=k+2;
ndinti=z+5;for j=1:ndinti;for i=1:k;
    x(j*k+i)=x(i)+j*m*pi;y(j*k+i)=y(i);
end;end;plot(x,y);axis equal;
xc=0;yc=0;l=1;
liminfteta=0;limsupteta=2.3*pi;
inc=2*pi/100+0.00001;
for teta=liminfteta:inc:limsupteta;
    for i=1:k*(ndinti+1)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)
        *cos(fi-teta-pi/2)-y(i)*sin(fi-teta-pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)
        *sin(fi-teta-pi/2)+y(i)*cos(fi-teta-pi/2);l=l+1;
    end;end;l=l-1;
plot(x1,y1,'k');hold on;axis equal
axis([- (rr+1.6*m1), (rr+1.6*m1), -
(rr+1.6*m1), (rr+1.6*m1)])
Title('REZULTATUL GRAFIC AL SIMULARII GENERARII
ROTII DE LANT PRIN RULARE')
xlabel('Excursia pe abscisa sistemului de
reprezentare 2D [mm]')
ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')
%rutina de determinare a profilului

```



```

%plot(absc,ord,'r','LineWidth',1.5);
%rutina pentru limitarea diametrului
dim1=size(absc);dim=dim1(2);razlim=rr+8;
for i=1:dim
    r(i)=sqrt(abs(absc(i))^2+abs(ord(i))^2);
if r(i)>=razlim;r(i)=razlim;else end
    if ord(i)==0;ord(i)=0.000000001;else end
        if absc(i)>0;
if ord(i)>0;tetha(i)=atan(abs(ord(i))/
abs(absc(i))); else end;else end
            if absc(i)<0;if ord(i)>0;tetha(i)=pi/2-
atan(abs(ord(i))/abs(absc(i)))+pi/2;
else end;;else end
                if absc(i)<0;if ord(i)<0;
tetha(i)=atan(abs(ord(i))/abs(absc(i)))+pi;else
end; else end
                    if absc(i)>0;if ord(i)<0;
tetha(i)=pi/2-
atan(abs(ord(i))/abs(absc(i)))+3*pi/2; else
end;else end
end
for i=1:dim;
abscl(i)=r(i)*cos(tetha(i));
ordl(i)=r(i)*sin(tetha(i));
end
plot(abscl,ordl,'b','LineWidth',1.5)

```

Anexa 15**PROGRAM SIMURLAN1**

```

clear all;close all;m=10;z=17;%a=110;
fi=90*pi/180;m1=m;
rr=0.5*m*(z);dep=0*m;k=1;inc=pi/30;cons=1;
for i=pi:inc:2*pi;
x(k)=10*cos(i*cons);
y(k)=10*sin(i*cons)+dep;k=k+1;end;
for i=pi:-inc:0;
x(k)=15.7+5.70796*cos(i*cons);
y(k)=5.70796*sin(i*cons)+dep;k=k+1;end;k=k-1;
ndinti=z+5;
for j=1:ndinti;for i=1:k;
    x(j*k+i)=x(i)+j*m*pi;y(j*k+i)=y(i);
end
end
y(1)=20.7264;y(j*k+i)=20.7264;
plot(x,y);axis equal;
xc=0;yc=0;
l=1;
liminf-teta=0;limsup-teta=2.3*pi;inc=2*pi/100+0.00
001;
for teta=liminf-teta:inc:limsup-teta;
    for i=1:k*(ndinti+1)
        x1(l)=xc+rr*cos(fi-teta)+(x(i)-
rr*teta)*cos(fi-teta-pi/2)-y(i)*sin(fi-teta-
pi/2);
        y1(l)=yc+rr*sin(fi-teta)+(x(i)-
rr*teta)*sin(fi-teta-pi/2)+y(i)*cos(fi-teta-
pi/2);l=l+1;
    end
end;l=l-1;

```

```

plot(x1,y1,'k');hold on;axis equal
axis([- (rr+1.6*m1), (rr+1.6*m1), -
(rr+1.6*m1), (rr+1.6*m1)])
xlabel('Excursia pe abscisa sistemului de
reprezentare 2D [mm]')
ylabel('Excursia pe ordonata sistemului de
reprezentare 2D [mm]')
%rutina de determinare a profilului
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=yc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
            if intx==x1(j);if inty==y1(j);
if dist<mindist;

```



```

absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
%plot(absc,ord,'r','LineWidth',1.5);
%rutina pentru limitarea diametrului
diml=size(absc);dim=diml(2);razlim=rr+8;
for i=1:dim
    r(i)=sqrt(abs(absc(i))^2+abs(ord(i))^2);
if r(i)>=razlim;r(i)=razlim;else end
    if ord(i)==0;ord(i)=0.000000001;else end
    if absc(i)>0;
if
ord(i)>0;tetha(i)=atan(abs(ord(i))/abs(absc(i)))
;
else end;else end
        if absc(i)<0;if ord(i)>0;
tetha(i)=pi/2-
atan(abs(ord(i))/abs(absc(i)))+pi/2;
else end;else end
            if absc(i)<0;

```

```
if ord(i)<0;
tetha(i)=atan(abs(ord(i))/abs(absc(i)))+pi;
else end;else end
        if absc(i)>0;
if ord(i)<0;
tetha(i)=pi/2-
atan(abs(ord(i))/abs(absc(i)))+3*pi/2;
else end;
else end
end
for i=1:dim;
absc1(i)=r(i)*cos(tetha(i));
ord1(i)=r(i)*sin(tetha(i));
end
plot(absc1,ord1,'k','LineWidth',1.5);axis equal
```

Anexa 16**PROGRAM fig36**

```

close all;clear all;tic;
nincr=100;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
limunghi1=215*pi/180;limunghi2=limunghi1+(101.33
34*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=22.71035;consty1=-
16.23102;
    limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=-0.16718;consty1=12.81608;

limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
k=k+1;
end
x(k)=0.20356;y(k)=10.87329;num=k;
plot(x,y,'k','LineWidth',2);axis equal;grid
axis([-5.5 0.5 9 13.2])

```

Anexa 17**PROGRAM fig37**

```

close all;clear all;tic;
r=13.01717;a=100;b=100;
nincr=100;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
limunghi1=215*pi/180;limunghi2=limunghi1+(101.33
34*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=22.71035;consty1=-
16.23102;
limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);
    k=k+1;
end
for i=0:nincr;constx1=-0.16718;consty1=12.81608;
limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
    k=k+1;
end
x(k)=0.20356;y(k)=10.87329;num=k;
%x1=x;y1=y;
for i=1:num;

```

```
        unghi(i)=pi+atan(y(i)/x(i));
        if x(i)>0;unghi(i)=atan(y(i)/x(i));else
end
        raza(i)=sqrt(x(i)^2+y(i)^2);
    end
    clear x;clear y;
k=1;
for j=0:11;
    for i=1:num;
        x(k)=raza(i)*cos(unghi(i)-pi/6*j);
        y(k)=raza(i)*sin(unghi(i)-pi/6*j);
        k=k+1;
    end
end
plot(x,y,'k','LineWidth',1.5);axis equal;grid
axis([-13.5 13.5 -13.5 13.5])
```

Anexa 18**PROGRAM DETPROFB**

```

close all;clear all;tic;
r=13.01717;a=100;b=0;nincr=100;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
limunghi1=215*pi/180;limunghi2=limunghi1+(101.33
34*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=22.71035;consty1=-
16.23102;
limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=-0.16718;consty1=12.81608;
limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
k=k+1;
end
x(k)=0.20356;y(k)=10.87329;num=k;
%x1=x;y1=y;
for i=1:num;
    unghi(i)=pi+atan(y(i)/x(i));

```

```

        if x(i)>0;unghi(i)=atan(y(i)/x(i));else
end
        raza(i)=sqrt(x(i)^2+y(i)^2);
    end
clear x;clear y;k=1;
for j=0:11;
    for i=1:num;
        x(k)=raza(i)*cos(unghi(i)-pi/6*j);
        y(k)=raza(i)*sin(unghi(i)-pi/6*j);
        k=k+1;
    end
end
%plot(x,y);axis equal;xdes=x;ydes=y;
k=k-1;
l=1;
x=-x;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
plot(x1,y1,'k');hold on;axis equal;hold on
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=-20:inc2:100;miny=1000;ics=0;

```

```

        abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;
                else end;else end;else end
                    if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'k','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
%axis([0,30,-15,15])
toc

```


Anexa 19**PROGRAM CLICHET1**

```

close all;clear all;tic;
r=13.01717;a=100;b=100;nincr=30;k=1;
for i=0:nincr;constx1=-4.68695;consty1=9.91983;
limunghi1=215*pi/180;limunghi2=limunghi1+(101.33
34*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+0.7*cos(limunghi1+i*incrunghi);
    y(k)=consty1+0.7*sin(limunghi1+i*incrunghi);
k=k+1;
end
for i=0:nincr;constx1=22.71035;consty1=-
16.23102;
limunghi1=136.3334*pi/180;limunghi2=limunghi1-
(8.1094*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+37.17452*cos(limunghi1-
i*incrunghi);
    y(k)=consty1+37.17452*sin(limunghi1-
i*incrunghi);
    k=k+1;
end
for i=0:nincr;constx1=-0.16718;consty1=12.81608;

limunghi1=128.224*pi/180;limunghi2=limunghi1-
(123.224*pi/180);
    incrunghi=(limunghi1-limunghi2)/nincr;
    x(k)=constx1+0.2*cos(limunghi1-i*incrunghi);
    y(k)=consty1+0.2*sin(limunghi1-i*incrunghi);
    k=k+1;
end
x(k)=0.20356;y(k)=10.87329;num=k;
%x1=x;y1=y;
for i=1:num;

```

```

        unghi(i)=pi+atan(y(i)/x(i));
        if x(i)>0;unghi(i)=atan(y(i)/x(i));else
end
        raza(i)=sqrt(x(i)^2+y(i)^2);
end
clear x;clear y;
k=1;
for j=0:11;
    for i=1:num;
        x(k)=raza(i)*cos(unghi(i)-pi/6*j);
        y(k)=raza(i)*sin(unghi(i)-pi/6*j);
        k=k+1;
    end
end
plot(x,y);axis equal
xdes=x;ydes=y;
k=k-1;
l=1;
x=-x;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;
plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;

```

```

end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=-20:inc2:80;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;
                else end;else end;else end
            if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+15.86;y=ord;rr=r;fi=pi/2;xc=0;yc=0;
liminfteta=0;limsupteta=3*pi;inc=2*pi/300;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
        end
        plot(x1,y1,'k');hold on;axis equal
    end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])
plot(xdes,ydes,'k','LineWidth',1.5)
toc

```

Anexa 20**PROGRAM CLICHET2**

```

close all;clear all;tic;
r=13.2;a=100;b=100;
x(1)=-4.7553;y(1)=8.7970;
x(2)=0;y(2)=13.2;
x(3)=0.2803;y(3)=9.9961;k=3;num=3;
%x1=x;y1=y;
    for i=1:num;
        unghi(i)=atan(y(i)/x(i));
        if
x(i)<0;unghi(i)=pi+atan(y(i)/x(i));else end
        raza(i)=sqrt(x(i)^2+y(i)^2);
        end
        clear x;clear y;
k=1;
for j=0:11;
    for i=1:num;
        x(k)=raza(i)*cos(unghi(i)-pi/6*j);
        y(k)=raza(i)*sin(unghi(i)-pi/6*j);
        k=k+1;
    end
end
plot(x,y);axis equal
xdes=x;ydes=y;
k=k-1;
l=1;
x=-x;
for i=0:3*pi/100:3*pi;u=-i;
for j=1:k;
    x1(l)=a-r*i+x(j)*cos(u)+y(j)*sin(u);
    y1(l)=b-x(j)*sin(u)+y(j)*cos(u);l=l+1;
end
end
l=l-1;

```

```

plot(x1,y1,'b');hold on;axis equal
maxx=-100;minx=100;
for i=1:l;
    if x1(i)>maxx;maxx=x1(i);else end;
    if x1(i)<minx;minx=x1(i);else end
    x2(i)=x1(l-i+1);y2(i)=y1(l-i+1);end
clear x1;clear y1;x1=x2;y1=y2;k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;
inc2=(maxx-minx)/6000;l=1;minminy=10000;
for i=-20:inc2:80;miny=1000;ics=0;
    abs(l)=i;
    for j=1:k-1;
        if i>x1(j);
if i<x1(j+1);int=m(j)*i+n(j);int;
            if int<miny;ord(l)=int;
miny=int;
                else end;else end;else end
                    if i<x1(j);if i>x1(j+1);int=m(j)*i+n(j);
int;
if int<miny;ord(l)=int;
miny=int;
else end;else end;else end
end;l=l+1;
if miny<minminy;minminy=miny;
else end
end
dim1=size(ord);dim=dim1(2);
plot(abs,ord,'r','LineWidth',1.5);
axis equal
ord=minminy-ord;
%plot(abs,ord,'k','LineWidth',1.5)
clear x1;clear y1;clear x;clear y;
x=abs+17.5;y=ord;rr=r;fi=pi/2;xc=0;yc=0;

```

```
liminfteta=0;limsupteta=3*pi;inc=2*pi/300;
for teta=liminfteta:inc:limsupteta;
    for i=1:dim
x1(i)=xc+rr*cos(fi-teta)+(x(i)-rr*teta)*cos(fi-
teta-pi/2)-y(i)*sin(fi-teta-pi/2);
y1(i)=yc+rr*sin(fi-teta)+(x(i)-rr*teta)*sin(fi-
teta-pi/2)+y(i)*cos(fi-teta-pi/2);
    end
    plot(x1,y1,'k');hold on;axis equal
end
axis([(xc-1.2*r),(xc+1.2*r),(yc-
1.2*r),(yc+1.2*r)])
plot(xdes,ydes,'k','LineWidth',1.5)
toc
```

Anexa 21**PROGRAM TRIUNGHI2**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=3;%nlat este numarul de lobi al sculei
%cu nlob=3 cercul baza si cercul rulanta au aceeasi
raza
rrul=50;rbaz=rrul*nlob/3;xc=0;yc=0;
x(1)=-43.3013;y(1)=-25;
x(2)=43.3013;y(2)=-25;
x(3)=0;y(3)=50;
x(4)=-43.3013;y(4)=-25;
k=4;
incr=2*pi/(500*nlob);
alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+3)/3*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);

y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
x1a=x1;y1a=y1;
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;

```

```

    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;

```



```
        else end
    end;
    l=l+1;
end
plot(absc,ord,'k','LineWidth',1.5);axis
equal;hold on

plot(x1a,y1a,'k','LineWidth',1.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
xcerc(l)=(rbaz+rrul)*cos(alfal)+rrul*cos(i);ycer
c(l)=(rbaz+rrul)*sin(alfal)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
```

Anexa 21 a**PROGRAM PATRAT2**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=3;%nlob este numarul de lobi al sculei
%cu nlob=4 cercul baza si cercul rulanta au
aceeasi raza
r=10.6066*2^0.5;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
k=5;
incr=2*pi/(500*nlob);
alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+4)/4*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
x1a=x1;y1a=y1;
clear x1;clear y1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI

```



```
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
plot(absc,ord,'k','LineWidth',1.5);axis
equal;hold on

plot(x1a,y1a,'k','LineWidth',1.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
xcerc(l)=(rbaz+rrul)*cos(alfa1)+rrul*cos(i);ycer
c(l)=(rbaz+rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
```

Anexa 22**PROGRAM TRIUNGHI3**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=1;%nlob este numarul de laturi al sculei
rrul=50;rbaz=rrul*nlob/3;xc=0;yc=0;
x(1)=-43.3013;y(1)=-25;
x(2)=43.3013;y(2)=-25;
x(3)=0;y(3)=50;
x(4)=-43.3013;y(4)=-25;
k=4;
incr=2*pi/(500);
alfa=0;beta=0;l=1;
for i=0:incr:2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+3)/3*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
end

```



```

%close all;plot(absc,ord,'k','LineWidth',1.5);
axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rbaz+rrul)/rrul*i;
        x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
        y(j)*sin(beta1);
        y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
        )*sin(beta1);
        %x2(1)=x1(j);y2(1)=y1(j);
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',1);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',2);
l=1;
for i=0:.01:2*pi

```

```
xcerc(l)=(rbaz+rrul)*cos(alfal)+rrul*cos(i);ycer  
c(l)=(rbaz+rrul)*sin(alfal)+rrul*sin(i);  
    l=l+1;  
end  
plot(xcerc,ycerc,'w','LineWidth',2);  
%RUTINA DE DETERMINARE A PROFILULUI piesei  
%axis([-rbaz rbaz -rbaz rbaz+10])  
toc
```


Anexa 22 a**PROGRAM PATRAT3**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=3;%nlob este numarul de lobi ai sculei
(profilului generator)
a=10.6066;r=a*2^0.5;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-a;y(1)=-a;x(2)=a;y(2)=-a;
x(3)=a;y(3)=a;x(4)=-a;y(4)=a;
x(5)=-a;y(5)=-a;
k=5;

incr=2*pi/(500);
alfa=0;beta=0;l=1;
for i=0:incr:2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(nlob+4)/4*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
)*sin(beta1);
        x2(1)=x1(j);y2(1)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end

```



```

l=l+1;
end
%close all;plot(absc,ord,'k','LineWidth',1.5);
axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rbaz+rrul)/rrul*i;
        x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
        y(j)*sin(beta1);
        y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
        )*sin(beta1);
        %x2(l)=x1(j);y2(l)=y1(j);
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',1);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',2);
l=1;
for i=0:.01:2*pi

```

```
xcerc(l)=(rbaz+rrul)*cos(alfal)+rrul*cos(i);ycer  
c(l)=(rbaz+rrul)*sin(alfal)+rrul*sin(i);  
    l=l+1;  
end  
plot(xcerc,ycerc,'w','LineWidth',2);  
%RUTINA DE DETERMINARE A PROFILULUI piesei  
%axis([-rbaz rbaz -rbaz rbaz+10])  
toc  
axis([-17 37 -17 32])
```

Anexa 23**PROGRAM SIMGENDANT1**

```

clear all;close all;tic
nlob=10;%nlob este numarul de dinti ai rotii de
generat;
rrul=250;rbaz=rrul*nlob/20;xc=0;yc=0;
load absc;load ord;clear x1;clear y1;
x=absc;y=ord;dim1=size(y);dim=dim1(2);
clear absc;clear ord
k=dim;
incr=2*pi/(150);
alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rbaz+rrul)/rrul*i;
x1(j)=(rbaz+rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
y1(j)=(rbaz+rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)
)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1a;clear y1a;
x1a=x1;y1a=y1;
clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',.5);hold on
axis equal
l=l-1;
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
end

```



```
plot(absc,ord,'k','LineWidth',2.5);axis
equal;hold on
plot(x1a,y1a,'k','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi

xcerc(l)=(rbaz+rrul)*cos(alfal)+rrul*cos(i);ycer
c(l)=(rbaz+rrul)*sin(alfal)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
toc
axis equal
axis([-180 600 -170 520])
```

Anexa 24**PROGRAM fig60**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI GENERATOR PRIN
RULARE CERC IN CERC
nlob=3;%nlob este numarul de lobi ai profilului
%generator,  $N_r$  în relația (23);
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
%plot(x,y,'k','LineWidth',1.5);hold on
k=5;
incr=2*pi/(50);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',0.5);hold on
axis equal;l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI GENERATOR

```



```

if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;l=l+1;
end
plot(absc,ord,'k','LineWidth',2.5);axis equal;
k=5;l=1;clear x1;clear y1;
for i=0.0:incr:0.0+incr;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
plot(x1,y1,'k','LineWidth',2.5)

```

Anexa 24 a**PROGRAM fig61**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI GENERATOR PRIN
RULARE CERC IN CERC
nlob=3;%nlob este numarul de lobi ai profilului
%generator, Nr în relația (23);
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
%plot(x,y,'k','LineWidth',1.5);hold on
k=5;
incr=2*pi/(50);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x2;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',0.5);
hold on;axis equal;l=l-1;

```

```

%ROUTINA DE DETERMINARE A PROFILULUI GENERATOR
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;

```

```

        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;l=l+1;
end
plot(absc,ord,'k','LineWidth',2.5);axis equal;
k=5;l=1;clear x1;clear y1;
for i=0.0:incr:0.0+incr;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
plot(x1,y1,'k','LineWidth',2.5);hold on
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rrul-
rbaz)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rrul-
rbaz)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);

```

Anexa 25**PROGRAM fig62**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI GENERATOR PRIN
RULARE CERC IN CERC
nlob=3;
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;

x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
plot(x,y,'k','LineWidth',1.5);hold on
k=5;
incr=2*pi/(50);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(1)=x1(j);y2(1)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;

```

```

%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;

```

```

        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
%r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(100);
alfa=0;beta=0;
l=1;
for i=0.2:incr:0.5*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rrul-
rbaz)/rbaz*i;
        x1(j)=(rrul-
rbaz)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rrul-
rbaz)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
    ;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1.5);hold on
axis equal
%plot(x1,y1,'w','LineWidth',2.5);
l=1;

```



```
for i=0:.01:2*pi
    xcerc(l)=rrul*cos(i);ycerc(l)=rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rrul-
rbaz)*cos(alfa1)+rbaz*cos(i);ycerc(l)=(rrul-
rbaz)*sin(alfa1)+rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
plot(x1,y1,'k','LineWidth',2.5);
toc
```

Anexa 26**PROGRAM PATRAT2**

```

clear all;close all;tic
nlob=3;
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
plot(x,y,'k','LineWidth',1.5);hold on
k=5;
%AICI INCEPE SINTEZA PROFILULUI GENERATOR PRIN
RULARE CERC IN CERC
incr=2*pi/(500);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;

```

```

%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;

```

```

        if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
%r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
alfa=0;beta=0;
l=1;
for i=0.1:incr:2.1*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rrul-
rbaz)/rbaz*i;
        x1(j)=(rrul-
rbaz)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rrul-
rbaz)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
    ;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1.5);hold on
axis equal
%plot(x1,y1,'w','LineWidth',2.5);
l=1;

```

```
for i=0:.01:2*pi
    xcerc(l)=rrul*cos(i);ycerc(l)=rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rrul-
rbaz)*cos(alfa1)+rbaz*cos(i);ycerc(l)=(rrul-
rbaz)*sin(alfa1)+rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
plot(x1,y1,'w','LineWidth',2.5);
toc
```

Anexa 27**PROGRAM CANELURAM3**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=4;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob
/6;xc=0;yc=0;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-
incdef:68*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-
incdef:52*cons;x(k)=10*cos(i);y(k)=10*sin(i);
k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=120*cons;for i=1:53;
x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;

```

```

    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis equal
k=321;l=1;
incr=2*pi/(100);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;

```

```

    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;

```



```

dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%ACI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;

```

```
end  
plot(xcerc,ycerc,'k','LineWidth',0.5);  
plot(x1,y1,'w','LineWidth',2.5);  
toc
```

Anexa 28**PROGRAM CANELURAM4**

```

clear all;close all;tic
nlob=4;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob
/6;xc=0;yc=0;
r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=9.6593;x(2)=-
2.5882;y(2)=11.5176;
x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;y(15)=11
.5176;
x(16)=2.5882;y(16)=9.6593;k=17;incdef=5.2871*con
s/10;k=17
for i=73.2871*cons:-
incdef:68*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-incdef:52*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
u=60*cons;for
i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=120*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=180*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);

```

```

y(k)=-x(i)*sin(u)+y(i)*cos(u);
k=k+1;end;u=240*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=300*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end
x(321)=x(1);y(321)=y(1);
plot(x,y);axis equal;k=321
incr=2*pi/(100);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);

```

```

        yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
        mi=(yb-ya)/dif;ni=ya-xa*mi;
        for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
            intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);

```

```

clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',1);hold on;axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
plot(x1,y1,'w','LineWidth',2.5);toc

```

Anexa 29**PROGRAM SIMGENDANT2**

```

clear all;close all;tic
nlob=21;%nlob este numarul de dinti ai sculei de
mortezat;
rrul=375;rbaz=rrul*nlob/30;xc=0;yc=0;
load absc;load ord;clear x1;clear y1;
x=absc;y=ord;xroata=x;yroata=y;
dim1=size(y);dim=dim1(2);x(dim+1)=1000;y(dim+1)=
0;dim=dim+1;
clear absc;clear ord
k=dim;
incr=2*pi/(100);alfa=pi;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(-1)*(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(-1)*(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));

```

```

        if dif==0;dif=0.00000000001;else end
        m(k)=(y1(i+1)-y1(i))/dif;
        n(k)=y1(i)-m(k)*x1(i);
        k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist<mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist<mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist<mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end

```



```

end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
%r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rrul-
rbaz)/rbaz*i;
        x1(j)=(rrul-
rbaz)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rrul-
rbaz)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1.5);hold on
axis equal
%plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rrul*cos(i);ycerc(l)=rrul*sin(i);
    l=l+1;
end
end

```

```
plot(xcerc,ycerc,'k','LineWidth',0.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rrul-
rbaz)*cos(alfal)+rbaz*cos(i);ycerc(l)=(rrul-
rbaz)*sin(alfal)+rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'k','LineWidth',0.5);
plot(x1,y1,'w','LineWidth',2.5);
toc
```

Anexa 30**PROGRAM fig74**

```

clear all;close all;tic;nlob=5;
r=10.6066*2^0.5;%r=15;
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
xroata=x;yroata=y;k=5;
incr=2*pi/(50);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin
(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',0.5);hold on;
axis equal;l=l-1;k=1;

```

```

%RUTINA DE DETERMINARE A PROFILULUI
SCULEI
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);k=k+1;
end; k=k-1;inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=-1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-
yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if
inty==y1(j+1);
if dist>mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else
end;else end;
                if intx==x1(j);if inty==y1(j);
if dist>mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;

```

```

        if intx>x1(j+1);if intx<x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
        else end
end;l=l+1;
end
plot(absc,ord,'k','LineWidth',2.5);axis
equal;
%IN ABSC SI ORD SUNT COORDONATELE
PUNCTELOR DE PE TAISUL SCULEI
l=1;k=5;clear x2;clear y2;
for i=0.2:incr:0.2;
    for j=1:k;
        alfa1=alfa+i;beta1=beta+(rrul-
rbaz)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin
(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',2.5);hold on
axis equal;axis([-20 17 -19 19])

```

Anexa 31**PROGRAM PATRAT3**

```

clear all;close all;tic
nlob=5;
r=10.6066*2^0.5
rrul=r;rbaz=rrul*nlob/4;xc=0;yc=0;
x(1)=-10.6066;y(1)=-10.6066;
x(2)=10.6066;y(2)=-10.6066;
x(3)=10.6066;y(3)=10.6066;
x(4)=-10.6066;y(4)=10.6066;
x(5)=-10.6066;y(5)=-10.6066;
xroata=x;yroata=y;
k=5;
incr=2*pi/(250);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI

```



```

if dist>mindist;absc(1)=intx;ord(1)=inty;
mindist=dist;else end;else end;else end;
    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%ACI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR GENEREAZA PROFIL PE
PIESA
%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);

```



```
        l=l+1;
    end
    plot(xcerc,ycerc,'w','LineWidth',1.5);
    l=1;
    for i=0:.01:2*pi
        xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
        l=l+1;
    end
    plot(xcerc,ycerc,'w','LineWidth',1.5);toc
```

Anexa 32**PROGRAM CANELURAM5**

```

clear all;close all;tic
%AICI INCEPE SINTEZA PROFILULUI DE SCULA PRIN
RULARE CERC PE CERC
nlob=8;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob
/6;xc=0;yc=0;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=8.6198;x(2)=-2.5882;
y(2)=11.5176;x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;
y(15)=11.5176;x(16)=2.5882;y(16)=8.6198;k=17;
incdef=5.2871*cons/10;k=17;
for i=73.2871*cons:-
incdef:68*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-
incdef:52*cons;x(k)=10*cos(i);y(k)=10*sin(i);
k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=9*cos(i);
y(k)=9*sin(i);k=k+1;end;u=60*cons;
for i=1:53;
x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=120*cons;for i=1:53;
x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=180*cons;for i=1:53;

```

```

    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=240*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;u=300*cons;for i=1:53;
    x(k)=x(i)*cos(u)+y(i)*sin(u);y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;
end;x(321)=x(1);y(321)=y(1);plot(x,y);axis equal
k=321;l=1;
xroata=x;yroata=y;
incr=2*pi/(200);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;

```

```

for i=0:inc:2.4*pi;mindist=-1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);
    yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
    mi=(yb-ya)/dif;ni=ya-xa*mi;
    for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist>mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist>mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI

```

```

r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1; x=absc;
y=ord;k=dim;
%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',0.5);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1.5);
toc

```

Anexa 33**PROGRAM CANELURAM6**

```

clear all;close all;tic
nlob=8;
r=12;a=100;b=0;cons=pi/180;rrul=r;rbaz=rrul*nlob
/6;xc=0;yc=0;
r=12;a=100;b=0;cons=pi/180;
incdef=22.4422*cons/10;k=1;
x(1)=-2.5882;y(1)=9.6593;x(2)=-
2.5882;y(2)=11.5176;
x(3)=-2.3352;y(3)=11.7706;k=4;
for i=101.2211*cons:-
incdef:78.7789*cons;x(k)=12*cos(i);
y(k)=12*sin(i);k=k+1;end;k=k-1;
x(14)=2.3352;y(14)=11.7706;x(15)=2.5882;y(15)=11
.5176;
x(16)=2.5882;y(16)=9.6593;k=17;incdef=5.2871*con
s/10;k=17
for i=73.2871*cons:-
incdef:68*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=16*cons/16;
for i=68*cons:-incdef:52*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;incdef=5.2871*cons/10;
for i=52*cons:-
incdef:46.7129*cons;x(k)=10*cos(i);
y(k)=10*sin(i);k=k+1;end;
u=60*cons;for
i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=120*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=180*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);

```

```

y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=240*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-
x(i)*sin(u)+y(i)*cos(u);k=k+1;end;u=300*cons;
for i=1:53;x(k)=x(i)*cos(u)+y(i)*sin(u);
y(k)=-x(i)*sin(u)+y(i)*cos(u);k=k+1;end
x(321)=x(1);y(321)=y(1);
plot(x,y);axis equal;k=321;xroata=x;yroata=y;
incr=2*pi/(200);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end
    m(k)=(y1(i+1)-y1(i))/dif;
    n(k)=y1(i)-m(k)*x1(i);
    k=k+1;
end; k=k-1;k
inc=2*pi/10000;l=1;
for i=0:inc:2.4*pi;mindist=-1000;
    xa=xc;ya=xc;xb=xa+5*cos(i);

```

```

        yb=ya+5*sin(i);dif=xb-xa;if
dif==0;dif=0.00000000001;else end;
        mi=(yb-ya)/dif;ni=ya-xa*mi;
        for j=1:k-1; dif1=mi-m(j);
if dif1==0;dif1=0.00000000001;else end;
            intx=(n(j)-ni)/dif1;
inty=intx*m(j)+n(j);
dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
if dist>mindist;absc(l)=intx;
ord(l)=inty;mindist=dist;else end;else end;else
end;
                if intx==x1(j);if inty==y1(j);
if dist>mindist;
absc(l)=intx;ord(l)=inty;mindist=dist;
else end;else end;else end;
                    if intx<x1(j+1);if intx>x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                        if intx>x1(j+1);if intx<x1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                            if inty<y1(j+1);if inty>y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                if inty>y1(j+1);if inty<y1(j);
if dist>mindist;absc(l)=intx;ord(l)=inty;
mindist=dist;else end;else end;else end;
                                    else end
end;
l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;

```



```

%incr=2*pi/(30*nlob);alfa=0;beta=0;
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',1);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1.5);
toc

```

Anexa 34**PROGRAM SIMGENDANT3**

```

clear all;close all;tic
nlob=29;
rrul=250;rbaz=rrul*nlob/20;xc=0;yc=0;
load absc;load ord;clear x1;clear y1;
x=absc;y=ord;dim1=size(y);dim=dim1(2);
xroata=x;yroata=y;
clear absc;clear ord
k=dim;
incr=2*pi/(100);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%RUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));
    if dif==0;dif=0.00000000001;else end

```



```

l=l+1;
end
%plot(absc,ord,'k','LineWidth',1.5);axis equal;
%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE
PE TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC
PRIN CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-
rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-
y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1)
;
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',.5);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1);
l=1;

```

```
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
    rrul)*cos(alfal)+rrul*cos(i);ycerc(l)=(rbaz-
    rrul)*sin(alfal)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1);
toc
```

Anexa 35**PROGRAM FIG87**

```

close all;clear all;
r=27.5;a=100;b=100;nincr=30;k=1;
%GENERARE DINTE 1 %Generare arc de flanc stanga;
nincr=30;k=1;
for i=0:nincr;constx1=21.5315;consty1=6.6331;
    limunghi1=158.7645*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de cap;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=75*pi/180;limunghi2=limunghi1-
(15*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+25*cos(limunghi1+i*incrunghi);
    y(k)=consty1+25*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de flanc dreapta;
for i=0:nincr;constx1=-10.5348;consty1=19.9154;
    limunghi1=4.308*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end

```

```
%Generare arc de picior;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=45*pi/180;limunghi2=limunghi1-
(45*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+15*cos(limunghi1+i*incrunghi);
    y(k)=consty1+15*sin(limunghi1+i*incrunghi);
k=k+1;
end
k=k-1;
plot(x,y);hold on;axis equal
l=1;
for j=0:3;
    alfa=cos(-j*pi/2);beta=sin(-j*pi/2);
    for i=1:k;
        x1(l)=x(i)*alfa-y(i)*beta;
        y1(l)=y(i)*alfa+x(i)*beta;l=l+1
    end
end
l=l-1;clear x;clear y;x=x1;y=y1;k=1;l=1;
plot(x,y,'k','LineWidth',1.5);axis([-30 30 -26 26])
```

Anexa 36**PROGRAM SIMGENDANT4**

```

clear all;close all;tic;nlob=6;
rrul=25;rbaz=rrul*nlob/4;xc=0;yc=0;
nincr=30;k=1;
for i=0:nincr;constx1=21.5315;consty1=6.6331;
    limunghi1=158.7645*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de cap;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=75*pi/180;limunghi2=limunghi1-
(15*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+25*cos(limunghi1+i*incrunghi);
    y(k)=consty1+25*sin(limunghi1+i*incrunghi);
k=k+1;
end
%Generare arc de flanc dreapta;
for i=0:nincr;constx1=-10.5348;consty1=19.9154;
    limunghi1=4.308*pi/180;limunghi2=limunghi1-
(28.0725*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;

x(k)=constx1+23.10003*cos(limunghi1+i*incrunghi);

y(k)=consty1+23.10003*sin(limunghi1+i*incrunghi);
k=k+1;
end

```



```

%Generare arc de picior;
for i=0:nincr;constx1=0;consty1=0;
    limunghi1=45*pi/180;limunghi2=limunghi1-
(45*pi/180);
    incrunghi=(limunghi2-limunghi1)/nincr;
    x(k)=constx1+15*cos(limunghi1+i*incrunghi);
    y(k)=consty1+15*sin(limunghi1+i*incrunghi);
k=k+1;
end
k=k-1;
%plot(x,y);hold on;axis equal
    l=1;
for j=0:3;
    alfa=cos(-j*pi/2);beta=sin(-j*pi/2);
for i=1:k;
    x1(l)=x(i)*alfa-y(i)*beta;
    y1(l)=y(i)*alfa+x(i)*beta;
    l=l+1;
end
end
l=l-1;clear x;clear y; x=x1;y=y1;
xroata=x;yroata=y;
k=1;l=1;
incr=2*pi/(500);alfa=0;beta=0;l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1);
        x2(l)=x1(j);y2(l)=y1(j);l=l+1;
    end
end
clear x1;clear x1;x1=x2;y1=y2;
%plot(x1,y1,'k','LineWidth',1);hold on
axis equal
l=l-1;
%ROUTINA DE DETERMINARE A PROFILULUI SCULEI
k=1;
for i=1:l-1;
    dif=(x1(i+1)-x1(i));

```

```

        if dif==0;dif=0.00000000001;else end
        m(k)=(y1(i+1)-y1(i))/dif;
        n(k)=y1(i)-m(k)*x1(i);
        k=k+1;
    end; k=k-1;k
    inc=2*pi/10000;l=1;
    for i=0:inc:2.4*pi;mindist=-1000;
        xa=xc;ya=xc;xb=xa+5*cos(i);
        yb=ya+5*sin(i);dif=xb-xa;if
    dif==0;dif=0.00000000001;else end;
        mi=(yb-ya)/dif;ni=ya-xa*mi;
        for j=1:k-1; dif1=mi-m(j);
    if dif1==0;dif1=0.00000000001;else end;
        intx=(n(j)-ni)/dif1;
    inty=intx*m(j)+n(j);
    dist=(abs(intx-xa)^2+abs(inty-ya)^2)^0.5;
        dist1=(abs(intx-xb)^2+abs(inty-yb)^2)^0.5;
        if dist1<dist;
            if intx==x1(j+1);if inty==y1(j+1);
    if dist>mindist;absc(l)=intx;
    ord(l)=inty;mindist=dist;else end;else end;else end;
        if intx==x1(j);if inty==y1(j);
    if dist>mindist;
    absc(l)=intx;ord(l)=inty;mindist=dist;
    else end;else end;else end;
        if intx<x1(j+1);if intx>x1(j);
    if dist>mindist;absc(l)=intx;ord(l)=inty;
    mindist=dist;else end;else end;else end;
        if intx>x1(j+1);if intx<x1(j);
    if dist>mindist;absc(l)=intx;ord(l)=inty;
    mindist=dist;else end;else end;else end;
        if inty<y1(j+1);if inty>y1(j);
    if dist>mindist;absc(l)=intx;ord(l)=inty;
    mindist=dist;else end;else end;else end;
        if inty>y1(j+1);if inty<y1(j);
    if dist>mindist;absc(l)=intx;ord(l)=inty;
    mindist=dist;else end;else end;else end;
        else end
    end;
    l=l+1;
    end
    %plot(absc,ord,'k','LineWidth',1.5);axis equal;

```

```

%IN ABSC SI ORD SUNT COORDONATELE PUNCTELOR DE PE
TAISUL SCULEI
r1=rrul;r2=rbaz;rbaz=r1;rrul=r2;
dim1=size(ord);dim=dim1(2);
clear x;clear y;clear x1;clear y1;
x=absc;y=ord;k=dim;
%AICI INCEPE SECTIUNEA DE RULARE CERC PE CERC PRIN
CARE PROFILUL GENERATOR
%GENEREAZA PROFIL PE PIESA
incr=2*pi/(200);
l=1;
for i=0.2:incr:2.2*pi;
    for j=1:k;
        alfa1=alfa+i;beta1=beta-(rbaz-rrul)/rrul*i;
        x1(j)=(rbaz-
rrul)*cos(alfa1)+x(j)*cos(beta1)-y(j)*sin(beta1);
        y1(j)=(rbaz-
rrul)*sin(alfa1)+y(j)*cos(beta1)+x(j)*sin(beta1);
        l=l+1;
    end
    plot(x1,y1,'k','LineWidth',0.5);hold on
end
%clear x1;clear x1;x1=x2;y1=y2;
plot(x1,y1,'k','LineWidth',.5);hold on
axis equal
plot(x1,y1,'w','LineWidth',2.5);
l=1;
for i=0:.01:2*pi
    xcerc(l)=rbaz*cos(i);ycerc(l)=rbaz*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1);
l=1;
for i=0:.01:2*pi
    xcerc(l)=(rbaz-
rrul)*cos(alfa1)+rrul*cos(i);ycerc(l)=(rbaz-
rrul)*sin(alfa1)+rrul*sin(i);
    l=l+1;
end
plot(xcerc,ycerc,'w','LineWidth',1);
plot(xroata,yroata,'k','LineWidth',2);
toc

```