

TEZĂ DE DOCTORAT
ROBOTIZAREA OPERAȚIILOR DE ANALIZĂ
METALOGRAFICĂ
ANEXE

CONDUCĂTOR ȘTIINȚIFIC

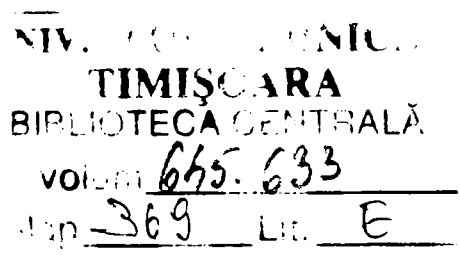
Prof. Dr. Ing.Dr.H.C.Mult.

FRANCISC VILIAM KOVÁCS

DOCTORAND

Șef lucrări ing.

TIBERIU VESSELENYI



2005

Cuprins

ANEXA 5.1.....	1
ANEXA 5.2.1.....	2
ANEXA 5.2.2.....	10
ANEXA 5.3.1.....	20
ANEXA 5.3.2.....	36
ANEXA 7.1.....	43
ANEXA 7.2.....	44
ANEXA 7.3.....	45
ANEXA 8.1.....	47
ANEXA 8.2.....	48
ANEXA 8.3.....	55
ANEXA 8.4.....	63
ANEXA 8.5.....	65
ANEXA 8.6.....	69
ANEXA 8.7.....	73
ANEXA 8.8.....	78
ANEXA 8.9.....	80
ANEXA 8.10.....	85
ANEXA 8.11.....	89
ANEXA 8.12.....	95
ANEXA 9.1.....	97
ANEXA 9.2.....	110
ANEXA 9.3.....	112
ANEXA 9.4.....	121
ANEXA 9.5.....	123
ANEXA 9.6.....	133
ANEXA 9.7.....	136

Subrutine utilizate în programele de aplicație celulă.

```
% Subrutina "incarcare"  
#incarcare(robot,postdelucru)  
execute/go(robot,postdelucru)  
execute/load(robot,postdelucru)  
  
% Subrutina "prelucrare asistata de robot"  
#prel_ar(robot,postdelucru)  
execute/go(robot,postdelucru)  
execute/down(robot,postdelucru)  
execute/assist(robot,postdelucru)  
execute/job(void,postdelucru)  
execute/up(robot,postdelucru)  
  
% Selectarea reactivului utilizat  
#select_reac(reactiv,date)  
execute/eval(reactiv,date)  
if reactiv.result is 'nital'  
    execute/go(robot_v5,nital)  
    execute/down(robot_v5,nital)  
    execute/assist(robot_v5,nital)  
    execute/job(void,nital)  
    execute/up(robot_v5,nital)  
endif  
if reactiv.result is 'persulfat'  
    execute/go(robot_v5,persulfat)  
    execute/down(robot_v5,persulfat)  
    execute/assist(robot_v5,persulfat)  
    execute/job(void,persulfat)  
    execute/up(robot_v5,persulfat)  
endif
```

Definirea obiectelor virtuale și a matricilor „metode”.

1. Clasa „Stocare”(STO)

Proprietățile obiectelor clasei STO

Nume proprietate	Valoare	Tip	Observații
name	<nume_stocator>	String	denumire
id	<id_stocator>	Int	identificator
status	0,1	Int	stare
probe_id	1,...,10	Int	identificator eșantion
probe_code	-	String	cod extern eșantion
probe_path	-	Int	itinerar eșantion
probe_template	-	Int	șablon eșantion
probe_priority	-	Int	prioritate inițială eșantion

nume_stocator	id_stocator
stocator01	1
stocator11	2
stocator12	3
stocator12	4
stocator21	5
stocator22	6
stocator23	7
stocator31	8

Matrice „metoda”, specificator de execuție „go”

Nume obiect STO	robot_h5	robot_v6
stocator01	10	0
stocator11	11	10
stocator12	12	11
stocator13	13	12
stocator21	0	13
stocator22	0	14
stocator23	0	15
stocator31	0	16

Matrice „metoda”, specificator de execuție „load”

Nume obiect STO	robot_h5	robot_v6
stocator01	20	0
stocator11	21	20
stocator12	22	21
stocator13	23	22
stocator21	0	23
stocator22	0	24
stocator23	0	25
stocator31	0	26

Matrice „metoda”, specificator de execuție „unload”

Nume obiect STO	robot_h5	robot_v6
stocator01	30	0
stocator11	31	30
stocator12	32	31
stocator13	33	32
stocator21	0	33
stocator22	0	34
stocator23	0	35
stocator31	0	36

2. Clasa „Prelucrare Asistată de Robot”(PAR)

Proprietățile obiectelor clasei PAR

Nume proprietate	Valoare	Tip	Observații
name	<nume_par>	String	denumire
id	<id_par>	Int	identificator
status	0,1	Int	stare
job_result	<rezultat>	String	rezultat procesare
probe_id	-	Int	identificator eșantion
probe_code	-	String	cod extern eșantion
probe_path	-	Int	itinerar eșantion
probe_template	-	Int	șablon eșantion
probe_priority	-	Int	prioritate inițială eșantion

nume_par	id_par	rezultat
analiza_supr	1	ok / slefuire_ebos / slefuire_fina
slefuire_ebos	2	-
slefuire_fina	3	-
spalare_slef	4	-
spalare_reac	5	-
uscare_slef	6	-
uscare_reac	7	-
nital	8	-
persulfat	9	-
asteptare_h5	10	-
asteptare_v6	11	-

Matrice „metoda”, specificator de execuție „go”

Nume obiect PAR	robot_h5	robot_v6
analiza_supr	40	0
slefuire_ebos	41	0
slefuire_fina	42	0
spalare_slef	43	0
spalare_reac	0	40
uscare_slef	44	0
uscare_reac	0	41
nital	0	42
persulfat	0	43
asteptare_h5	45	44
asteptare_v5	46	0

Matrice „metoda”, specificator de execuție „up”

Nume obiect PAR	robot_h5	robot_v6
analiza_supr	50	0
slefuire_ebos	51	0
slefuire_fina	52	0
spalare_slef	53	0
spalare_reac	0	50
uscare_slef	54	0
uscare_reac	0	51
nital	0	52
persulfat	0	53
asteptare_h5	55	54
asteptare_v5	56	0

Matrice „metoda”, specificator de execuție „down”

Nume obiect PAR	robot_h5	robot_v6
analiza_supr	60	0
slefuire_ebos	61	0
slefuire_fina	62	0
spalare_slef	63	0
spalare_reac	0	60
uscare_slef	64	0
uscare_reac	0	61
nital	0	62
persulfat	0	63
asteptare_h5	65	64
asteptare_v5	66	0

Matrice „metoda”, specificator de execuție „assist” pentru robot_h5

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	105	105	105
slefuire_ebos	105	115	110
slefuire_fina	110	110	105
spalare_slef	105	105	105
spalare_reac	0	0	0
uscare_slef	105	105	105
uscare_reac	0	0	0
nital	0	0	0
persulfat	0	0	0
asteptare_h5	105	105	105

Matrice „temp”, specificator de execuție „assist” pentru robot h5

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	5	5	5
slefuire_ebos	5	15	10
slefuire_fina	10	10	5
spalare_slef	5	5	5
spalare_reac	0	0	0
uscare_slef	5	5	5
uscare_reac	0	0	0
nital	0	0	0
persulfat	0	0	0
asteptare_h5	5	5	5

Matrice „metoda”, specificator de execuție „assist” pentru robot v6

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	0	0	0
slefuire_ebos	0	0	0
slefuire_fina	0	0	0
spalare_slef	0	0	0
spalare_reac	107	107	107
uscare_slef	0	0	0
uscare_reac	107	107	107
nital	112	115	0
persulfat	0	0	115
asteptare_v5	105	105	105

Matrice „temp”, specificator de execuție „assist” pentru robot v6

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	0	0	0
slefuire_ebos	0	0	0
slefuire_fina	0	0	0
spalare_slef	0	0	0
spalare_reac	7	7	7
uscare_slef	0	0	0
uscare_reac	7	7	7
nital	12	15	0
persulfat	0	0	15
asteptare_v5	5	5	5

Matrice „metoda”, specificator de execuție „job”

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	1	1	1
slefuire_ebos	0	0	0
slefuire_fina	0	0	0
spalare_slef	0	0	0
spalare_reac	0	0	0
uscare_slef	0	0	0
uscare_reac	0	0	0
nital	0	0	0
persulfat	0	0	0
asteptare_h5	0	0	0
asteptare_v5	0	0	0

Matrice „temp”, specificator de execuție „job”

Nume obiect PAR	Șablon 1	Șablon 2	Șablon 3
analiza_supr	5	5	5
slefuire_ebos	0	0	0
slefuire_fina	0	0	0
spalare_slef	0	0	0
spalare_reac	0	0	0
uscare_slef	0	0	0
uscare_reac	0	0	0
nital	0	0	0
persulfat	0	0	0
asteptare_h5	0	0	0
asteptare_v5	0	0	0

3. Clasa „Prelucrare Independentă de Robot”(PIR)

Proprietățile obiectelor clasei PIR

Nume proprietate	Valoare	Tip	Observații
name	<nume_pir>	String	denumire
id	<id_pir>	Int	identificator
load_status	0,1	Int	stare de încărcare
active_status	0,1	Int	stare de activitate
job_result	<rezultat>*	String	rezultat procesare
probe_id	-	Int	identificator eşantion
probe_code	-	String	cod extern eşantion
probe_path	-	Int	itinerar eşantion
probe_template	0	Int	şablon eşantion
probe_priority	0	Int	prioritate inițială eşantion

nume_pir	id_pir
analiza_stru	1

* Rezultatul procesării este specificat în funcție de algoritmul de analiză metalografică.

Matrice „metoda”, specificator de execuție „go”

Nume obiect PIR	Robot 1	Robot 2
analiza_stru	0	70

Matrice „metoda”, specificator de execuție „load”

Nume obiect PIR	Robot 1	Robot 2
analiza_stru	0	80

Matrice „metoda”, specificator de execuție „unload”

Nume obiect PIR	Robot 1	Robot 2
analiza_stru	0	90

Matrice „metoda”, specificator de execuție „job”

Nume obiect PIR	Şablon 1	Şablon 2	Şablon 3
analiza_stru	1	2	3

Matrice „temp”, specificator de execuție „job”

Nume obiect PIR	Şablon 1	Şablon 2	Şablon 3
analiza_stru	20	30	50

4. Clasa „Robot”(ROBOT)

Proprietățile obiectelor clasei ROBOT

Nume proprietate	Valoare	Tip	Observații
name	<nume_robot>	String	denumire
id	<id_robot>	Int	identificator
status	0,1	Int	stare robot
griper	0,1	Int	stare griper
probe_id	-	Int	identificator eșantion
probe_code	-	String	cod extern eșantion
probe_path	-	Int	itinerar eșantion
probe_template	-	Int	șablon eșantion
probe_priority	-	Int	prioritate inițială eșantion

nume_robot	id_robot
robot_h5	1
robot_v6	2

5. Clasa “Decizie”(DEC)

Proprietățile obiectelor clasei DEC

Nume proprietate	Valoare	Tip	Observații
name	<nume_dec>	String	denumire
id	<id_dec>	Int	identificator
status	0,1	Int	stare
result	<rezultat>	String	rezultat procesare
data	<date>	String	date

nume_dec	id_dec	rezultat	date
optimizare	1	stocator11 / stocator12 / stocator13 / stocator21 / stocator22 / stocator23	date1
reactiv	2	nital / persulfat	date2

Module program robot H5.

1.Clasa „Stocare”(STO)

1.1.Stocator01

1.1.1. Specificator de execuție „go” - MPR :10

10,
 1,,,38200,,,*****.*****,,*****.*****.....40,,
 2,,36200.45100.....*****.*****,,*****.*****.....40..
 3,,,,,*****.*****,,*****.*****.....40,5.

1.1.2. Specificator de execuție „load” - MPR :20

20,
 1,,,1200,,,*****.*****,,*****.*****.....40,,
 2,,,,,19000,*****.*****,,*****.*****.....40..
 3,,,38200,,,*****.*****,,*****.*****.....40..
 4,,,,,*****.*****,,*****.*****.....40,5.

1.1.3. Specificator de execuție „unload” - MPR :30

30,
 1,,,,,19000,*****.*****,,*****.*****.....40..
 2,,,1200,,,*****.*****,,*****.*****.....40..
 3,,,,,29000,*****.*****,,*****.*****.....40..
 4,,,38200,,,*****.*****,,*****.*****.....40..
 5,,,,,*****.*****,,*****.*****.....40,5.

1.2.Stocator11

1.2.1. Specificator de execuție „go” - MPR :11

11.

1,,...38200,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
2,,41100.11700,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
3,,..... ***** , ***** ,,, ***** , ***** ,,,,,,40.5,

1.2.2. Specificator de execuție „load” - MPR :21

21.

1,,...30000,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
2,,.....19000, ***** , ***** ,,, ***** , ***** ,,,,,,40,,
3,,...38200,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
4,,..... ***** , ***** ,,, ***** , ***** ,,,,,,40.5,

1.2.3. Specificator de execuție „unload” - MPR :31

31.

1,,.....19000, ***** , ***** ,,, ***** , ***** ,,,,,,40,,
2,,...30000,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
3,,.....29000, ***** , ***** ,,, ***** , ***** ,,,,,,40,,
4,,...38200,,... ***** , ***** ,,, ***** , ***** ,,,,,,40,,
5,,..... ***** , ***** ,,, ***** , ***** ,,,,,,40.5,

1.3.Stocator12

1.3.1. Specificator de execuție „go” - MPR :12

12,

1,,,38200,,,***** ,***** ,,,***** ,***** ,,,,,,40,,
2,,44700,11100,,,***** ,***** ,***** ,***** ,,,,,,40,,
3,,,,,***** ,***** ,,,***** ,***** ,,,,,,40,5,

1.3.2. Specificator de execuție „load” - MPR :22

22,

1,,,30000,,,***** ,***** ,,,***** ,***** ,,,,,,40,,
2,,,,,19000,***** ,***** ,***** ,***** ,,,,,,40,,
3,,,38200,,,***** ,***** ,,,***** ,***** ,,,,,,40,,
4,,,,,***** ,***** ,,,***** ,***** ,,,,,,40,5,

1.3.3. Specificator de execuție „unload” - MPR :32

32,

1,,,,,19000,***** ,***** ,,,***** ,***** ,,,,,,40,,
2,,,30000,,,***** ,***** ,,,***** ,***** ,,,,,,40,,
3,,,,,29000,***** ,***** ,,,***** ,***** ,,,,,,40,,
4,,,38200,,,***** ,***** ,***** ,***** ,,,,,,40,,
5,,,,,***** ,***** ,,,***** ,***** ,,,,,,40,5,

1.4.Stocator13

1.4.1. Specificator de execuție „go” - MPR :13

13.

1,,,38200,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
2,,49200,12500,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
3,,,,,***** ,***** ,,, ***** ,***** ,,,,,,40,5,

1.4.2. Specificator de execuție „load”- MPR :23

23,

1,,,30000,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
2,,,,,19000,***** ,***** ,,, ***** ,***** ,,,,,,40,,
3,,,38200,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
4,,,,,***** ,***** ,,, ***** ,***** ,,,,,,40,5,

1.4.3. Specificator de execuție „unload” - MPR :33

33.

1,,,,,19000,***** ,***** ,,, ***** ,***** ,,,,,,40,,
2,,,30000,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
3,,,,,29000,***** ,***** ,,, ***** ,***** ,,,,,,40,,
4,,,38200,,,***** ,***** ,,, ***** ,***** ,,,,,,40,,
5,,,,,***** ,***** ,,, ***** ,***** ,,,,,,40,5,

2. Clasa „Prelucrare Asistată de Robot”(PAR)

2.1. Analiza_supr

2.1.1. Specificator de execuție “go” – MPR : 40

40,

1....38200....***** ,***** ,...***** ,*****40..
2..49400.45200....***** ,***** ,...***** ,*****40..
3.....***** ,***** ,...***** ,*****40.5,

2.1.2. Specificator de execuție “up” – MPR : 50

50,

1....38200....***** ,***** ,...***** ,*****40..
2.....***** ,***** ,...***** ,*****40.5,

2.1.3. Specificator de execuție “down” – MPR : 60

60,

1...32700....***** ,***** ,...***** ,*****40..
2.....***** ,***** ,...***** ,*****40.5,

2.1.4. Specificator de execuție “assist”

2.1.4.1. Șablon 1 – MPR : 105

105,

1.....***** ,***** ,...***** ,***** ,1.2...,40..
2.....***** ,***** ,...***** ,*****40.5,

2.1.4.2. Șablon 2 – MPR : 105

2.1.4.3. Șablon 3 – MPR : 105

2.2. Slefuire_ebos

2.1.1. Specificator de execuție “go” – MPR : 41

41,

1....38200,....,*****,,*****,,*****,,*****,,*****,,.....40,,
2..34000,61600,....,*****,,*****,,*****,,*****,,*****,,.....40,,
3.....,*****,,*****,,*****,,*****,,*****,,.....40,5.

2.1.2. Specificator de execuție “up” – MPR : 51

51,

1....38200,....,*****,,*****,,*****,,*****,,*****,,.....40,,
2.....,*****,,*****,,*****,,*****,,*****,,.....40,5.

2.1.3. Specificator de execuție “down” – MPR : 61

61,

1....,16300,....,*****,,*****,,*****,,*****,,*****,,.....40,,
2.....,*****,,*****,,*****,,*****,,*****,,.....40,5.

2.1.4. Specificator de execuție “assist”

2.1.4.1.Șablon 1 – MPR : 105

105,

1.....,*****,,*****,,*****,,*****,,*****,,1.2,....40,,
2.....,*****,,*****,,*****,,*****,,*****,,.....40,5,

2.1.4.2.Șablon 2 – MPR : 115

2.1.4.3.Șablon 3 – MPR : 110

2.3. Slefuire_fina

2.1.1. Specificator de execuție “go” – MPR : 42

42,

1....38200,...***** ,***** ,...***** ,*****40..
2..29800.48500,...***** ,***** ,...***** ,*****40..
3.....***** ,***** ,...***** ,*****40.5.

2.1.2. Specificator de execuție “up” – MPR : 52

52,

1....38200,...***** ,***** ,...***** ,*****40..
2.....***** ,***** ,...***** ,*****40.5.

2.1.3. Specificator de execuție “down” – MPR : 62

62,

1....16300,...***** ,***** ,...***** ,*****40..
2.....***** ,***** ,...***** ,*****40.5.

2.1.4. Specificator de execuție “assist”

2.1.4.1.Șablon 1 – MPR : 110

105,

1.....***** ,***** ,...***** ,***** ,1,2....40..
2.....***** ,***** ,...***** ,***** ,.....40.5.

2.1.4.2.Șablon 2 – MPR : 110

2.1.4.3.Șablon 3 – MPR : 105

2.4. Spalare_slef

2.1.1. Specificator de execuție "go" – MPR : 43

43.

1....38200,....,*****,,*****,,*****,,*****,,*****,,40,,
2..39700.44000,....,*****,,*****,,*****,,*****,,*****,,40,,
3.....,*****,,*****,,*****,,*****,,*****,,40.5,

2.1.2. Specificator de execuție "up" – MPR : 53

53.

1....38200,....,*****,,*****,,*****,,*****,,*****,,40,,
2.....,*****,,*****,,*****,,*****,,*****,,40.5,

2.1.3. Specificator de execuție "down" – MPR : 63

63.

1,,,10000,....,*****,,*****,,*****,,*****,,*****,,40,,
2,,,,,,*****,,*****,,*****,,*****,,*****,,40.5,

2.1.4. Specificator de execuție "assist"

2.1.4.1.Șablon 1 – MPR : 105

105.

1,,,,,,*****,,*****,,*****,,*****,,*****,,1,2,,,40,,
2,,,,,,*****,,*****,,*****,,*****,,*****,,40.5,

2.1.4.2.Șablon 2 – MPR : 105

2.1.4.3.Șablon 3 – MPR : 105

2.6. Uscare_slef

2.1.1. Specificator de execuție “go” – MPR : 44

44.

1....38200....***** ,***** ,,, ***** ,*****40..
2..40700.39000.....***** ,***** ,***** ,*****40..
3.....***** ,***** ,,, ***** ,*****40.5.

2.1.2. Specificator de execuție “up” – MPR : 54

54.

1....38200....***** ,***** ,,, ***** ,*****40..
2.....***** ,***** ,***** ,*****40.5.

2.1.3. Specificator de execuție “down” – MPR : 64

64.

1....10000....***** ,***** ,,, ***** ,*****40..
2.....***** ,***** ,***** ,*****40.5.

2.1.4. Specificator de execuție “assist”

2.1.4.1.Șablon 1 – MPR : 105

105.

1.....***** ,***** ,,, ***** ,***** .1.2....40..
2.....***** ,***** ,,, ***** ,*****40.5.

2.1.4.2.Șablon 2 – MPR : 105

2.1.4.3.Șablon 3 – MPR : 105

2.10. Asteptare_h5

2.1.1. Specificator de execuție “go” – MPR : 45

45,

1...38200,... ***** ,***** ,,, ***** ,*****40..
2..58000.6700..... ***** ,***** ,***** ,*****40..
3..... ***** ,***** ,,, ***** ,*****40.5.

2.1.2. Specificator de execuție “up” – MPR : 55

55,

1...38200,... ***** ,***** ,,, ***** ,*****40..
2..... ***** ,***** ,,, ***** ,*****40.5,

2.1.3. Specificator de execuție “down” – MPR : 65

65,

1...38200,... ***** ,***** ,,, ***** ,*****40..
2..... ***** ,***** ,,, ***** ,*****40.5.

2.1.4. Specificator de execuție “assist”

2.1.4.1.Șablon 1 – MPR : 105

105,

1..... ***** ,***** ,,, ***** ,***** ,1.2,,,40..
2..... ***** ,***** ,,, ***** ,***** ,,,,,40.5,

2.1.4.2.Șablon 2 – MPR : 105

2.1.4.3.Șablon 3 – MPR : 105

Schema bloc a celulei flexibile CFLM1.

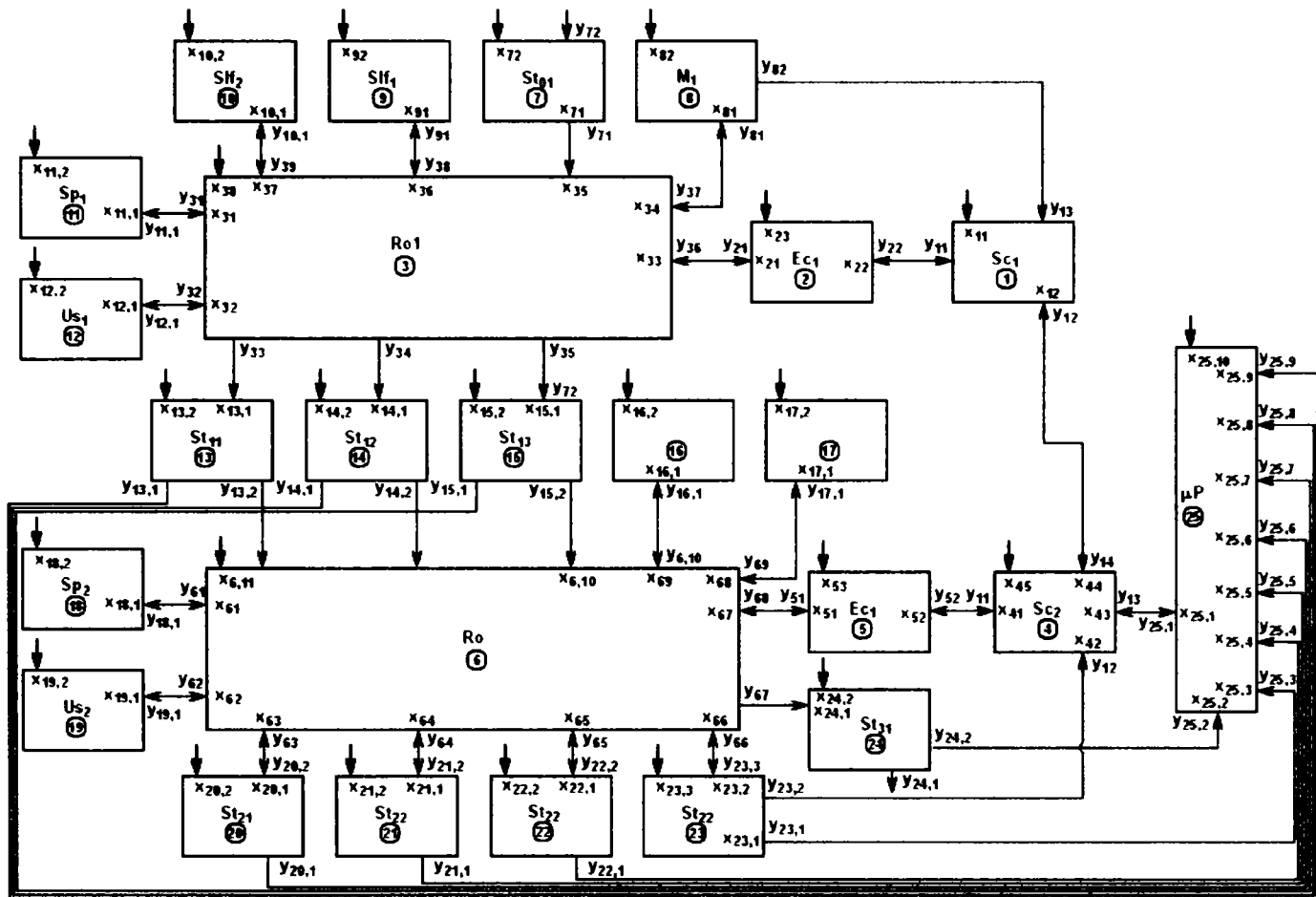


Fig.A.5.1.1 Schema bloc a celulei flexibile CFLM1.

Matricile de cuplare ale CFLM1.

$$\underline{K}_{12} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{21} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{14} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{ccc} 1 & 2 & 3 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{41} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{18} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{23} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \left[\begin{array}{ccccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{32} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{37} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{38} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{39} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} 1 \\ \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{array} \right] \end{array}$$

$$\underline{K}_{3,11} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \end{array}$$

$$\underline{K}_{3,12} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 0 \\ 2 & 1 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \end{array}$$

$$\underline{K}_{25,7} = \begin{array}{c|cc} i \backslash o & 1 & 2 \\ \hline 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \\ 4 & 0 & 0 \\ 5 & 0 & 0 \\ 6 & 0 & 0 \\ 7 & 0 & 0 \\ 8 & 0 & 0 \\ 9 & 0 & 1 \\ 10 & 0 & 0 \end{array}$$

$$\underline{K}_{83} = \begin{array}{c|cccccccc} i \backslash o & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\underline{K}_{93} = \begin{array}{c|cccccccc} i \backslash o & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\underline{K}_{10,3} = \begin{array}{c|cccccccc} i \backslash o & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\underline{K}_{11,3} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{12,3} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{13,3} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{14,3} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{15,3} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{54} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{ccc} 1 & 2 & 3 \end{array} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{45} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{cc} 1 & 2 \end{array} \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{K}_{4,25} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{ccc} 1 & 2 & 3 \end{array} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{65} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

⋮

$$\underline{K}_{s_6} = \begin{array}{c|cc} i \backslash o & 1 & 2 \\ \hline 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \\ 4 & 0 & 0 \\ 5 & 0 & 0 \\ 6 & 0 & 0 \\ 7 & 1 & 0 \\ 8 & 0 & 0 \\ 9 & 0 & 0 \\ 10 & 0 & 0 \\ 11 & 0 & 0 \\ 12 & 0 & 0 \\ 13 & 0 & 0 \end{array}$$

$$\underline{K}_{13.6} = \begin{array}{c|cc} i \backslash o & 1 & 2 \\ \hline 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \\ 4 & 0 & 0 \\ 5 & 0 & 0 \\ 6 & 0 & 0 \\ 7 & 0 & 0 \\ 8 & 0 & 0 \\ 9 & 0 & 0 \\ 10 & 0 & 0 \\ 11 & 0 & 0 \\ 12 & 0 & 1 \\ 13 & 0 & 0 \end{array}$$

$$\underline{K}_{14.6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{15.6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{16.6} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \\ 9 & 1 \\ 10 & 0 \\ 11 & 0 \\ 12 & 0 \\ 13 & 0 \end{array}$$

$$\underline{K}_{17.6} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 1 \\ 9 & 0 \\ 10 & 0 \\ 11 & 0 \\ 12 & 0 \\ 13 & 0 \end{array}$$

$$\underline{K}_{18,6} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \\ 9 & 0 \\ 10 & 0 \\ 11 & 0 \\ 12 & 0 \\ 13 & 0 \end{array}$$

$$\underline{K}_{19,6} = \begin{array}{c|c} i \backslash o & 1 \\ \hline 1 & 0 \\ 2 & 1 \\ 3 & 0 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \\ 9 & 0 \\ 10 & 0 \\ 11 & 0 \\ 12 & 0 \\ 13 & 0 \end{array}$$

$$\underline{K}_{20,6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{21,6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{22,6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{K}_{23,6} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{6,16} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{6,17} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{6,18} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K}_{6,19} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{6,20} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{6,21} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{6,22} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{6,23} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{6,24} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \end{array} \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{25,7} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} 1 \ 2 \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \end{array}$$

$$\underline{K}_{25,13} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} 1 \\ 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{25,14} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} 1 \\ 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{25,15} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} 1 \\ 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{25.20} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{K}_{25.21} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{K}_{25.22} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{K}_{25,23} = \begin{array}{c|ccc} i \backslash o & 1 & 2 & 3 \\ \hline 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 \\ 11 & 0 & 0 & 0 \end{array}$$

$$\underline{K}_{25,24} = \begin{array}{c|cc} i \backslash o & 1 & 2 \\ \hline 1 & 0 & 0 \\ 2 & 0 & 1 \\ 3 & 0 & 0 \\ 4 & 0 & 0 \\ 5 & 0 & 0 \\ 6 & 0 & 0 \\ 7 & 0 & 0 \\ 8 & 0 & 0 \\ 9 & 0 & 0 \\ 10 & 0 & 0 \\ 11 & 0 & 0 \end{array}$$

$$\underline{K}_{25,4} = \begin{array}{c|ccc} i \backslash o & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 \\ 11 & 0 & 0 & 0 \end{array}$$

$$\underline{K}_{23,25} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

$$\underline{K}_{4,25} = \begin{array}{c} i \backslash o \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{cc} 1 & 2 \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array}$$

:

Definirea elementelor modelării cu rețele Petri a CFLM1.

Tabel A.5.3.2.1.

Nr. Crt	Simbol	Tip	Semnificație	Caracteristici
1	P1	Poz.	Ro1 liber pentru.epruveta A	$Mo(P1)=1$
2	P2	Poz.	epruveta A pe St01	$Mo(P2)=1$
3	T1	Tr.	Ro1 preia epruveta A de pe St01	$d(T1)= 5$
4	P3	Poz.	Ro1 cu epruveta A la Slf1	$Mo(P3)=0$
5	P4	Poz.	St01 liber	$Mo(P4)=0$
6	T2	Tr.	Epruveta A pe St01	$T2= 60$
7	T3	Tr.	Ciclu de șlefuire epruveta A pe Slf1	$T3= 120$
8	P5	Poz.	Ro1 are epruveta A pentru Slf1	$Mo(P5)=0$
9	T4	Tr.	Ciclu de șlefuire epruveta A Slf2	$d(T4)= 80$
10	P6	Poz.	Ro1 are epruveta A pentru Slf2	$Mo(P6)=0$
11	T5	Tr.	Ro1 depune epruveta A pe St11	$d(T5)= 5$
12	T6	Tr.	Ro1 depune epruveta A pe St12	$d(T6)= 5$
13	T7	Tr.	Ro1 depune epruveta A pe St13	$d(T7)= 5$
14	P7	Poz.	Epruveta A pe St11	$Mo(P7)=0$
15	P8	Poz.	Epruveta A pe St12	$Mo(P8)=0$
16	P9	Poz.	Epruveta A pe St13	$Mo(P9)=0$
17	T8	Tr.	Confirmare : există epruveta A pe St11	$d(T8)=0$ (tr.log.)
18	T9	Tr.	Confirmare : există epruveta A pe St12	$d(T9)=0$ (tr.log.)
19	T10	Tr.	Confirmare : există epruveta A pe St13	$d(T10)=0$ (tr.log.)
20	P10	Poz.	Există epruveta A pe St11 (Cond. de ev.)	$Mo(P10)=0$
21	P11	Poz.	Există epruveta A pe St12 (Cond. de ev.)	$Mo(P11)=0$
22	P12	Poz.	Există epruveta A pe St13 (Cond. de ev.)	$Mo(P12)=0$
23	T11	Tr.	Ro2 preia epruveta A de pe St11	$d(T11)= 5$
24	T12	Tr.	Ro2 preia epruveta A de pe St12	$d(T12)= 5$
25	T13	Tr.	Ro2 preia epruveta A de pe St13	$d(T13)= 5$
26	P13	Poz.	Nu există. epruveta A pe St11	$Mo(P13)=1$
27	P14	Poz.	Nu există. epruveta A pe St12	$Mo(P14)=1$
28	P15	Poz.	Nu există. epruveta A pe St13	$Mo(P15)=1$
29	P16	Poz.	Ro2 are epruveta A de pe St11	$Mo(P16)=0$

30	P17	Poz.	Ro2 are epruveta A de pe St12	Mo(P17)=0
31	P18	Poz.	Ro2 are epruveta A de pe St13	Mo(P18)=0
32	T14	Poz.	Confirmare : există. epruveta A la Ro2 pe St11	d(T14)=0 (tr.log.)
33	T15	Poz.	Confirmare : există. epruveta A la Ro2 pe St12	d(T15)=0 (tr.log.)
34	T16	Poz.	Confirmare : există. epruveta A la Ro2 pe St13	d(T16)=0 (tr.log.)
35	T19	Poz.	Ro2 are epruveta A șlefuită	Mo(P19)=0
36	T17	Tr.	Ro2 are epruveta A pe R1	d(T17)= 10
37	P20	Poz.	Ro2 are epruveta A tratată cu R1	Mo(P20)=0
38	T18	Tr.	Ro2 spală epruveta A	d(T18)= 10
39	P21	Poz.	Ro2 are epruveta A spălată	Mo(P21)=0
40	T19	Tr.	Ro2 usucă epruveta A	d(T19)= 10
41	P22	Poz.	Ro2 are epruveta A uscată	Mo(P22)=0
42	T20	Tr.	Ro2 duce epruveta A direct la M2	d(T20)= 15
43	P23	Poz.	Epruveta A este la M2	Mo(P23)=0
44	T21	Tr.	Epruveta A este examinată	d(T21)= 60
45	P24	Poz.	Epruveta A examinată	Mo(P24)=0
46	T22	Tr.	Ro2 preia epruveta A	d(T22)= 10
47	P25	Poz.	Ro2 are epruveta A	Mo(P25)=0
48	T23	Tr.	Ro2 evacuează epruveta A	d(T23)= 15
49	P26	Poz.	Contorizare epruveta A	Mo(P26)=0
50	T24	Tr.	Confirmare : depunere epruveta A pe St2	Tr. log. d(T24)=0
51	P27	Poz.	Ro2. cu epruveta A pentru a o depune pe St2	Mo(P27)=0
52	P28	Poz.	St21 este liber pentru epruveta A	Mo(P28)=1
53	P29	Poz.	St22 este liber pentru epruveta A	Mo(P29)=1
54	P30	Poz.	St23 este liber pentru epruveta A	Mo(P30)=1
55	T25	Tr.	Ro2 depune epruveta A pe St21	d(T25)= 10
56	T26	Tr.	Ro2 depune epruveta A pe St22	d(T26)= 10
57	T27	Tr.	Ro2 depune epruveta A pe St23	d(T27)= 10
58	P31	Poz.	La St21 există epruveta A	Mo(P31)=0
59	P32	Poz.	La St22 există epruveta A	Mo(P32)=0
60	P33	Poz.	La St23 există epruveta A	Mo(P33)=0
61	T28	Tr.	Ro2 preia epruveta A de pe St21 pentru a o depune la M2	d(T28)= 10
62	T29	Tr.	Ro2 preia epruveta A de pe St22 pentru a o depune la M2	d(T29)= 10

63	T30	Tr.	Ro2 preia epruveta A de pe St23 pentru a o depune la M2	$d(T30)= 10$
64	P34	Poz.	Ro2 are epruveta A de pe St21	$Mo(P34)=0$
65	P35	Poz.	Ro2 are epruveta A de pe St22	$Mo(P35)=0$
66	P36	Poz.	Ro2 are epruveta A de pe St23	$Mo(P36)=0$
67	T31	Tr.	Ro2 depune epruveta A de pe St21 pe M2	$d(T31)= 5$
68	T32	Tr.	Ro2 depune epruveta A de pe St22 pe M2	$d(T32)= 5$
69	T33	Tr.	Ro2 depune epruveta A de pe St23 pe M2	$d(T33)= 5$
70	P37	Poz.	Epruveta A de pe St21 este la M2	$Mo(P37)=0$
71	P38	Poz.	Epruveta A de pe St22 este la M2	$Mo(P38)=0$
72	P39	Poz.	Epruveta A de pe St23 este la M2	$Mo(P39)=0$
73	T34	Tr.	Epruveta A de pe St21 este examinată la M2	$d(T34)= 60$
74	T35	Tr.	Epruveta A de pe St22 este examinată la M2	$d(T35)= 60$
75	T36	Tr.	Epruveta A de pe St22 este examinată la M2	$d(T36)= 60$
76	P40	Poz.	La M2 există epruveta A de pe St21 examinată	$Mo(P40)=0$
77	P41	Poz.	La M2 există epruveta A de pe St22 examinată	$Mo(P41)=0$
78	P42	Poz.	La M2 există epruveta A de pe St23 examinată	$Mo(P42)=0$
79	T37	Tr.	Ro2 preia epruveta A examinată	$d(T37)= 10$
80	T38	Tr.	Ro2 preia epruveta A examinată	$d(T38)= 10$
81	T39	Tr.	Ro2 preia epruveta A examinată	$d(T39)= 10$
82	P43	Poz.	Ro2 are epruveta A	$Mo(P43)=0$
83	P44	Poz.	Ro2 are epruveta A	$Mo(P44)=0$
84	P45	Poz.	Ro2 are epruveta A	$Mo(P45)=0$
85	T40	Tr.	Ro2 evacuează. epruveta A	$d(T40)= 15$
86	T41	Tr.	Ro2 evacuează. epruveta A	$d(T41)= 15$
87	T42	Tr.	Ro2 evacuează. epruveta A	$d(T42)= 15$
88	P46	Poz.	M2 este liber	$Mo(P46)=1$
89	P47	Poz.	Ro1 liber pentru epruveta B	$Mo(P47)=0$
90	P48	Poz.	Epruveta B pe St02	$Mo(P48)=1$
91	T43	Tr.	Ro1 preia epruveta B de pe St02	$d(T43)= 5$
92	P49	Poz.	Ro1 cu epruveta B la St1	$Mo(P49)=0$
93	P50	Poz.	St02 liber	$Mo(P50)=0$
94	T44	Tr.	Epruveta B pe St02	$d(T44)= 60$
95	T45	Tr.	Ciclu de șlefuire epruveta B pe Sf1	$d(T45)= 120$

96	P51	Poz.	Ro1 are epruveta B pe Slf1	Mo(P51)=0
97	T46	Tr.	Ciclu de șlefuire epruveta B pe Slf2	d(T46)= 80
98	P52	Poz.	Ro1 are epruvetaB pe Slf2	Mo(P52)=0
99	T47	Tr.	Ro1 depune epruveta B pe St11	d(T47)= 5
100	T48	Tr.	Ro1 depune epruveta B pe St12	d(T48)= 5
101	T49	Tr.	Ro1 depune epruvetaB pe St13	d(T49)= 5
102	P53	Poz.	Epruveta B pe St11	Mo(P53)=0
103	P54	Poz.	Epruveta B pe St12	Mo(P54)=0
104	P55	Poz.	Epruveta B pe St13	Mo(P55)=0
105	T50	Tr.	Confirmare : există epruveta B pe St11	d(T50)=0 (tr.log.)
106	T51	Tr.	Confirmare : există epruveta B pe St12	d(T51)=0 (tr.log.)
107	T52	Tr.	Confirmare : există epruvetaA pe St13	d(T52)=0 (tr.log.)
108	P56	Poz.	Există epruveta B pe St11 (Cond. de ev.)	Mo(P56)=0
109	P57	Poz.	Există epruveta B pe St12 (Cond. de ev.)	Mo(P57)=0
110	P58	Poz.	Există epruveta B pe St13 (Cond. de ev.)	Mo(P58)=0
111	T53	Tr.	Ro2 preia epruveta B de pe St11	d(T53)= 5
112	T54	Tr.	Ro2 preia epruveta B de pe St12	d(T54)= 5
113	T55	Tr.	Ro2 preia epruveta B de pe St13	d(T55)= 5
114	P59	Poz.	Nu există. epruveta B pe St11	Mo(P59)=1
115	P60	Poz.	Nu există. epruveta B pe St12	Mo(P60)=1
116	P61	Poz.	Nu există. epruveta B pe St13	Mo(P61)=1
117	P62	Poz.	Ro2 are epruveta B de pe St11	Mo(P62)=0
118	P63	Poz.	Ro2 are epruveta B de pe St12	Mo(P63)=0
119	P64	Poz.	Ro2 are epruveta B de pe St13	Mo(P64)=0
120	T56	Tr.	Confirmare : există. epruveta B la Ro2 pe St11	d(T56)=0 (tr.log.)
121	T57	Tr.	Confirmare : există. epruveta B la Ro2 pe St12	d(T57)=0 (tr.log.)
122	T58	Tr.	Confirmare : există. epruveta B la Ro2 pe St13	d(T58)=0 (tr.log.)
123	P65	Poz.	Ro2 are epruveta B șlefuită	Mo(P65)=0
124	T59	Tr.	Ro2 are epruveta B pe R1	d(T59)= 10
125	P66	Poz.	Ro2 are epruveta B tratată la R1	Mo(P66)=0
126	T60	Tr.	Ro2 spală epruveta B	d(T60)= 10
127	P67	Poz.	Ro2 are epruveta B spălată	Mo(P67)=0

128	T61	Tr.	Ro2 usucă epruveta B	$d(T61)= 10$
129	P68	Poz.	Ro2 are epruveta B uscată	$Mo(P68)=0$
130	T62	Tr.	Ro2 duce epruveta B direct la M2	$d(T62)= 15$
131	P69	Poz.	epruveta B este la M2	$Mo(P69)=0$
132	T63	Tr.	epruveta B este examinată	$d(T63)= 60$
133	P70	Poz.	epruveta B examinată	$Mo(P70)=0$
134	T64	Tr.	Ro2 preia epruveta B	$d(T64)= 10$
135	P71	Poz.	Ro2 are epruveta B	$Mo(P71)=0$
136	T65	Tr.	Ro2 ev epruveta B	$d(T65)= 15$
137	P72	Poz.	Contorizare epruveta B	$Mo(P72)=0$
138	P73	Tr.	Contorizare epruveta A + epruveta B	$Mo(P73)=0$
139	T66	Tr.	Confirmare : depunere epruveta B pe St2	Tr. log. $d(T66)=0$
140	P74	Poz.	Ro2. cu epruveta B pentru a o depune pe St2	$Mo(P74)=0$
141	P75	Poz.	St21 este liber pentru epruveta B	$Mo(P75)=1$
142	P76	Poz.	St22 este liber pentru epruveta B	$Mo(P76)=1$
143	P77	Poz.	St23 este liber pentru epruveta B	$Mo(P77)=1$
144	T67	Tr.	Ro2 depune epruveta B pe St21	$d(T67)= 10$
145	T68	Tr.	Ro2 depune epruveta B pe St22	$d(T68)= 10$
146	T69	Tr.	Ro2 depune epruveta B pe St23	$d(T69)= 10$
147	P78	Poz.	Pe St21 există epruveta B	$Mo(P78)=0$
148	P79	Poz.	Pe St22 există epruveta B	$Mo(P79)=0$
149	P80	Poz.	Pe St23 există epruveta B	$Mo(P80)=0$
150	T70	Tr.	Ro2 preia epruveta B de pe St21 pentru a o depune la M2	$d(T70)= 10$
151	T71	Tr.	Ro2 preia epruveta B de pe St22 pentru a o depune la M2	$d(T71)= 10$
152	T72	Tr.	Ro2 preia epruveta B de pe St23 pentru a o depune la M2	$d(T72)= 10$
153	P81	Poz.	Ro2 are epruveta B de pe St21	$Mo(P81)=0$
154	P82	Poz.	Ro2 are epruveta B de pe St22	$Mo(P82)=0$
155	P83	Poz.	Ro2 are epruveta B de pe St23	$Mo(P83)=0$
156	T73	Tr.	Ro2 depune epruveta B de pe St21 pe M2	$d(T73)= 5$
157	T74	Tr.	Ro2 depune epruveta B de pe St22 pe M2	$d(T74)= 5$
158	T75	Tr.	Ro2 depune epruveta B de pe St23 pe M2	$d(T75)= 5$
159	P84	Poz.	Epruveta B de pe St21 este la M2	$Mo(P84)=0$
160	P85	Poz.	Epruveta B de pe St22 este la M2	$Mo(P85)=0$

161	P86	Poz.	Epruveta B de pe St23 este la M2	Mo(P86)=0
162	T76	Tr.	Epruveta B de pe St21 este examinată	d(T76)= 60
163	T77	Tr.	Epruveta B de pe St22 este examinată	d(T77)= 60
164	T78	Tr.	Epruveta B de pe St22 este examinată	d(T78)= 60
165	P87	Poz.	La M2 există epruveta B de pe St21 examinată	Mo(P87)=0
166	P88	Poz.	La M2 există epruveta B de pe St22 examinată	Mo(P88)=0
167	P89	Poz.	La M2 există epruveta B de pe St23 examinată	Mo(P89)=0
168	T79	Tr.	Ro2 preia epruveta B examinată	d(T79)= 10
169	T80	Tr.	Ro2 preia epruveta B examinată	d(T80)= 10
170	T81	Tr.	Ro2 preia epruveta B examinată	d(T81)= 10
171	P90	Poz.	Ro2 are epruvetaB	Mo(P90)=0
172	P91	Poz.	Ro2 are epruvetaB	Mo(P91)=0
173	P92	Poz.	Ro2 are epruvetaB	Mo(P92)=0
174	T82	Tr.	Ro2 evacuează epruveta B	d(T82)= 15
175	T83	Tr.	Ro2 evacuează epruveta B	d(T83)= 15
176	T84	Tr.	Ro2 evacuează epruveta B	d(T84)= 15
177	P93	Poz.	există cel puțin un loc liber pe St2	Mo(P93)=1
178	P94	Poz.	Ro2 liber	Mo(P94)=0
179	T85	Tr.	Confirmare : Ro2 va duce epruveta A direct la M2	d(T85)= 5
180	P95	Poz.	Ro2 are epruveta A pentru M2	Mo(P95)=0
181	T86	Tr.	Confirmare : Ro2 va duce epruveta B direct la M2	d(T86)= 5
182	P96	Poz.	Ro2 are epruveta B pt M2	Mo(P96)=0

Notații utilizate în tabelul 1.

A, B – notarea epruvetelor;

Poz. – poziție;

Tr. – tranziție

Cond. de ev. – condiție de evaluare

Ro1 – robot 1;

Ro2 – robot 2;

St01 – Stocator 01;

St11 – Stocator 11;

St12 – Stocator 12;

St13 – Stocator 13;

St21 – Stocator 21;

St22 – Stocator 22;

St23 – Stocator 23;
St31 – Stocator 31;
Slf1 – șlefuire 1 (șlefuire de degroșare);
Slf2 – șlefuire2 (șlefuire de finisare);
R1 – reactiv 1 (nital);
R2 – reactiv 2 (persulfat);
M2 – microscop de analiză microstructură;
 T_j – tranziția „j”;
 $Mo(P_j)$ – Marcaj în poziția P_j , 0 – nu există marcaj, 1 – există marcaj;
 $d(T_j)$ - temporizarea asociată tranziției T_j .

Tabel A7.1.1. Parametrii sistemului pneumatic.

Notație	Descriere	Notație MATLAB	Valoare	Unitate de măsură
m	Masa ansamblului în mișcare	m	0,115	kg
S	Lungimea cursei	S	0,09	m
x	Poziție momentană punct carac.	x	variabil	m
P_1	Presiune în camera 1	P1	variabil	N/m^2
P_2	Presiune în camera 2	P2	variabil	N/m^2
A_1	Arie pistonului	A1	$2,85 \times 10^{-4}$	m^2
A_2	Arie pistonului – aria secțiunii tijei	A2	$2,34 \times 10^{-4}$	m^2
b	Coefficient de amortizare	b	necunoscut	$N \times s/m$
q_1	Debit masic în camera 1	q1	variabil	kg/s
q_2	Debit masic în camera 2	q2	variabil	kg/s
V_{1min}	Volum inițial în camera 1	V1min	$3,62 \times 10^{-6}$	m^3
V_{2min}	Volum inițial în camera 2	V2min	$2,96 \times 10^{-7}$	m^3
T	Temperatura aerului	T	294	$^{\circ}K$
A_{pd1}	Suprafața orificiului electrovalva 1	Apd1	$7,92 \times 10^{-6}$	m^2
A_{pd2}	Suprafața orificiului electrovalva 2	Apd2	$7,92 \times 10^{-6}$	m^2
γ	Căldura specifică	gama	1,4	-
R	Constanta gazului ideal	R	287	$J/kg^{\circ}K$
P_s	Presiunea de intrare aer comprimat	Ps	$2,07 \times 10^6$	N/m^2
P_e	Presiunea de ieșire aer comprimat	Pe	0	N/m^2
ρ	Densitatea aerului	ro	1,2	kg/m^3
T_{i1}	Timp de întârziere pentru em 1	Ti1	0,011	s
T_{i2}	Timp de întârziere pentru em 2	Ti2	0,011	s

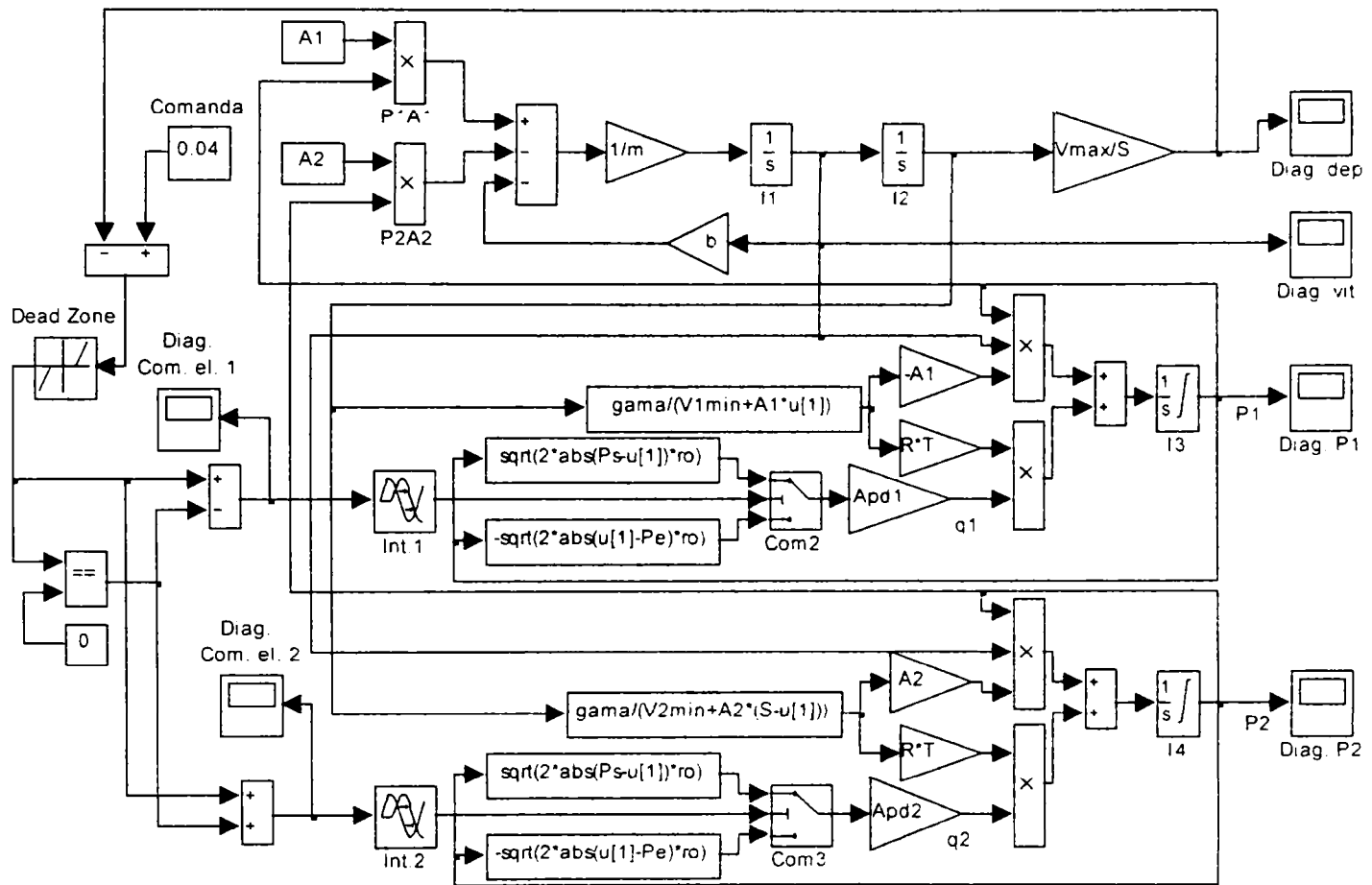


Fig.A.7.2.1. Modelul SIMULINK al sistemului cu comandă "on – off".

Rezultatele simulării pentru comanda “on-off”

Parametrii simulării :

- valoarea mărimii de comandă (referință) = 0,04 [m];
- $b = 180$ [Ns/m];
- $T_{i1} = T_{i2} = 0,011$ [s];
- “Dead zone” = -0,0017... +0,0017 [m];

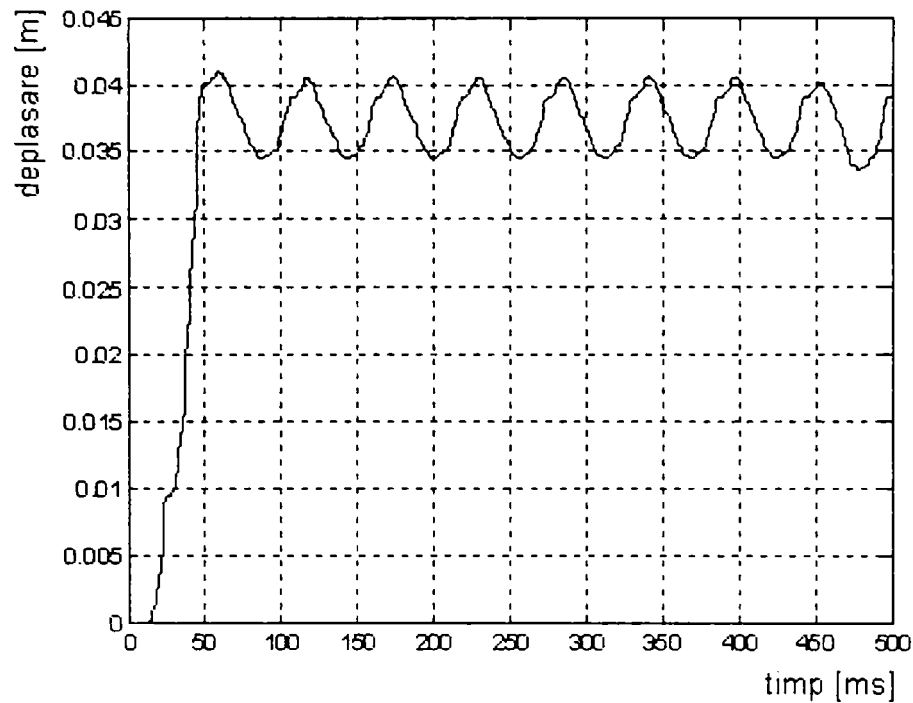


Fig. A.7.3.1. Diagrama deplasării pistonului pneumatic pentru comanda prezentată în diagramele din figurile A7.3.3 și A.7.3.4.

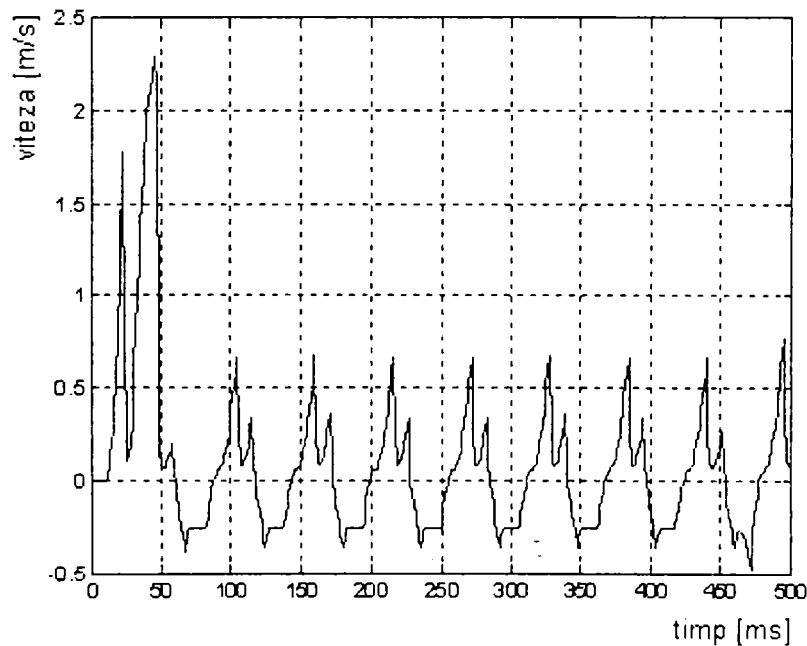


Fig. A.7.3.1. Diagrama deplasării pistonului pneumatic pentru

comanda prezentată în diagramele din figurile A7.3.3 și A.7.3.4.

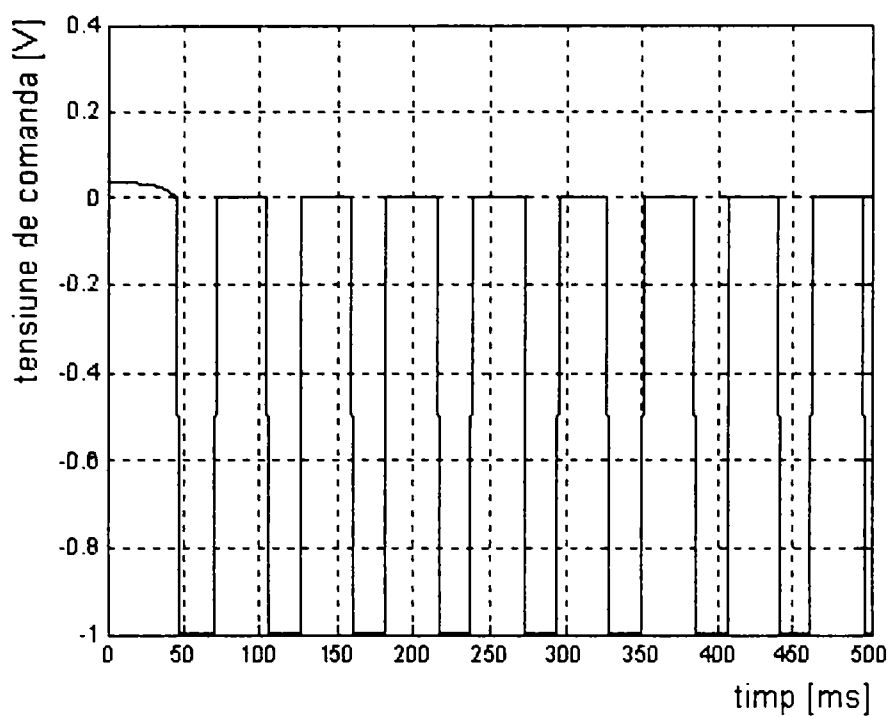


Fig.A.7.3.3. Semnalul de comandă pentru “el.1”.

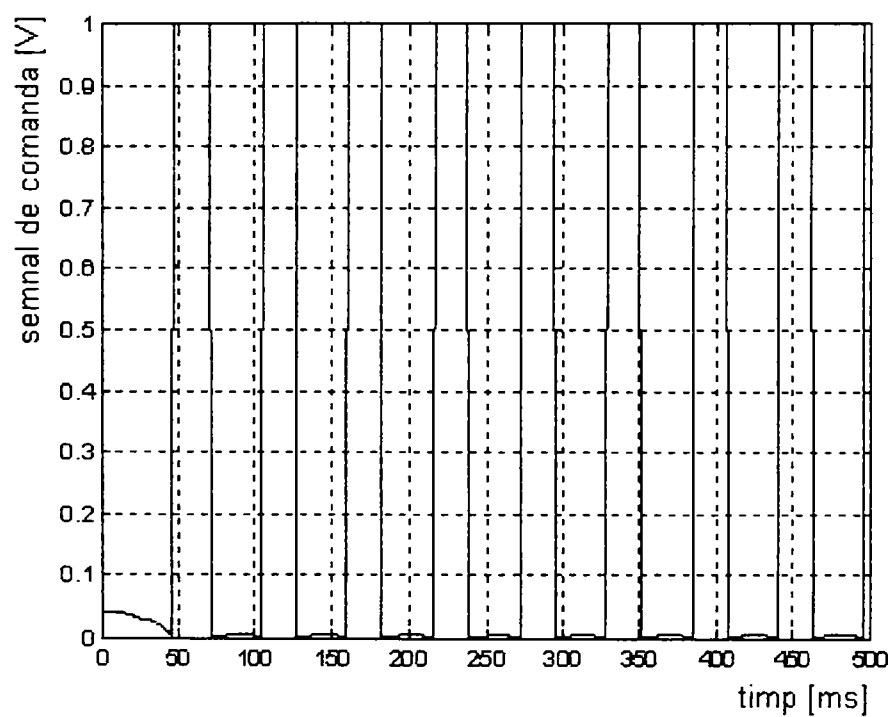





















Fig.A.7.3.4. Semnalul de comandă pentru “el.2”.

Tabel.1.

Cod	Rugozitate (R_a)
4	12,5...6,3
5	6,3...3,2
6	3,2...1,6
7	1,6...0,8
8	0,8...0,4
9	0,4...0,2
10	0,2...0,1
11	0,1...0,05
12	0,05...0,025
13	0,025...0,012

Tabel.2.

Denumire prelucrare	Cod rugozitate	Cod general	Simbol pentru reprezentare în diagrame
Frezare cilindrică	5	fc5	
	6	fc6	
	7	fc7	
Frezare frontală	5	ff5	
	6	ff6	
	7	ff7	
Rabotare plană	4	rab4	
	5	rab5	
	6	rab6	
	7	rab7	
Rectificare frontală	6	rf6	
	7	rf7	
	8	rf8	
Rectificare plană	6	rp6	
	7	rp7	
	8	rp8	
	9	rp9	
Netezire plană	10	np10	
	11	np11	
	12	np12	
	13	np13	

Program pentru preprocesarea imaginilor – “preproc1”

```

% Incarcare fisier imagine
[filename,pathname] = uigetfile('*.*.','Selectati fisierul imagine',300,100);
if filename ~=0
    %Citire imagine in matrice
    imagine = imread(strcat(pathname,filename));
    % Transformare din RGB in UINT8
    I = rgb2gray(imagine);
    [nr_linii, nr_coloane] = size(I);
    nr_se = 25;
    nr_rot = 1;
    filtru_gaus1;
    I1 = ordfilt2(Ib.5.ones(3,3));
    level1 = graythresh(I1);
    I2 = I1;
    % Transformare in bw
    BW1 = im2bw(I2,level1);
    % Eliminare gauri
    BW1a = imfill(BW1,'holes');
    qw1 = BW1a;
    qw2 = BW1a;
    qw1(floor(nr_linii/2),:) = 1;
    qw2(:,floor(nr_coloane/2)) = 1;
    qw1a = imcomplement(imfill(imcomplement(qw1),'holes'));
    qw2a = imcomplement(imfill(imcomplement(qw2),'holes'));
    BW2 = qw1a & qw2a;
    % Delimitare margini
    BW2(1:nr_se,1:nr_coloane) = 0;
    BW2(nr_linii-nr_se:nr_linii,1:nr_coloane) = 0;
    BW2(1:nr_linii,1:nr_se) = 0;
    BW2(1:nr_linii,nr_coloane-nr_se:nr_coloane) = 0;
    % Eliminarea neregularitatilor de contur ale imaginii
    se = strel('disk',nr_se);
    BW3 = imclose(BW2,se); bw = BW3;
    % Stabilire dreptunghi de incadrare maxim posibil
    j1 = 0;
    for k = 1:nr_coloane - 1
        a_det = 0; b_det = 0; c_det = 0; d_det = 0;
        for l = 1:nr_linii - 1
            if ((bw(l+1,k) - bw(l,k)) > 0)
                a_k = k+1;
                a_l = l+1;
                a_det = 1;
            end
            if ((bw(l+1,k) - bw(l,k)) < 0)
                b_k = k+1;
                b_l = l;
            end
        end
    end
end

```

```

        b_det = 1;
    end
end
if (a_det & b_det)
    for m = k+1:nr_coloane - 1
        if (bw(a_l,m+1) - bw(a_l,m) < 0)
            c_k = m;
            c_l = a_l;
            c_det = 1;
        end
        if (bw(b_l,m+1) - bw(b_l,m) < 0)
            d_k = m;
            d_l = b_l;
            d_det = 1;
        end
    end
end
if (a_det & b_det & c_det & d_det)
    if abs(c_k - a_k) > abs(d_k - b_k)
        lungime = abs(d_k - b_k);
        e_k = d_k;
        e_l = d_l;
    end
    if abs(c_k - a_k) <= abs(d_k - b_k)
        lungime = abs(c_k - a_k);
        e_k = c_k;
        e_l = c_l;
    end
    latime = abs(b_l - a_l);
    arie = latime*lungime;
    j1 = j1+1;
    rezultat(j1,:) = [a_k a_l lungime latime arie];
end
end
[max_arie max_linie] = max(rezultat(:,5));
a_k_max = rezultat(max_linie,1);
a_l_max = rezultat(max_linie,2);
lungime_max = rezultat(max_linie,3);
latime_max = rezultat(max_linie,4);
rezultat(:,5) = 0;
% Delimitare imagine dreptunghiulara
I3 = I2(a_l_max+5:a_l_max + latime_max-5,a_k_max+5:a_k_max + lungime_max-5);
% Eliminarea componentei de iluminare, si mentinerea componentei de reflexie
filtru_gaus;
% Afisare Imagine
figure; imshow(I3c);
filename
% Clonare imagine
[nr_linii, nr_coloane] = size(I3c);
clona = 256;

```

```

I1cl = (I3c(1:clona,1:clona));
I2cl = (I3c(nr_linii + 1 - clona:nr_linii,1:clona));
I3cl = (I3c(1:clona,nr_coloane + 1 - clona:nr_coloane));
I4cl = (I3c(nr_linii + 1 - clona:nr_linii,nr_coloane + 1 - clona:nr_coloane));

% Salvare clone
filename1 = strcat(pathname,filename(1:length(filename)-4),'_1');
filename2 = strcat(pathname,filename(1:length(filename)-4),'_2');
filename3 = strcat(pathname,filename(1:length(filename)-4),'_3');
filename4 = strcat(pathname,filename(1:length(filename)-4),'_4');
save(filename1,'I1cl');
imwrite(I1cl, strcat(filename1, '.bmp'), 'bmp');
save(filename2,'I2cl');
imwrite(I2cl, strcat(filename2, '.bmp'), 'bmp');
save(filename3,'I3cl');
imwrite(I3cl, strcat(filename3, '.bmp'), 'bmp');
save(filename4,'I4cl');
imwrite(I4cl, strcat(filename4, '.bmp'), 'bmp');
end

```

Subprogram “filtru_gaus”

```

% Generare filtru Gaussian cu dimensiunea a 1/5 din imaginea originala
dom_PSF = 51;
sigma_PSF = 49;
dom_1 = floor(dom_PSF/2);
dom_2 = dom_1+1;
PSF = fspecial('gaussian',dom_PSF,sigma_PSF);
% Filtrarea imaginii pentru generarea componentei de iluminare
CI1 = conv2(PSF,I3);
% Corectarea marginilor componentei de iluminare
CI2 = CI1(dom_1:end-dom_2,dom_1:end-dom_2); [n1fc n2fc] = size(CI2);
FC1 = conv2(PSF,ones(n1fc,n2fc));
FC2 = FC1(dom_1:end-dom_2,dom_1:end-dom_2);
CI3 = (double(CI2)./FC2)/256;
% Schimbarea domeniului imaginii initiale
I3a = double(I3);
% Extragerea componentei de iluminare din imaginea initiala
I3b = I3a./CI3;
I3c = I3b/max(max(I3b));

```

Subprogram “filtru_gaus1”

```

% Generare filtru Gaussian
dom_PSF = 51;
sigma_PSF = 49;
dom_1 = floor(dom_PSF/2);
dom_2 = dom_1+1;
PSF = fspecial('gaussian',dom_PSF,sigma_PSF);
% Filtrarea imaginii

```

```

C11 = conv2(PSF,1);
% figure: mesh(C11)
% Corectarea marginilor
CI2 = C11(dom_1:end-dom_2,dom_1:end-dom_2); [n1fc n2fc] = size(CI2);
FC1 = conv2(PSF,ones(n1fc,n2fc));
FC2 = FC1(dom_1:end-dom_2,dom_1:end-dom_2);
CI3 = (double(CI2)./FC2)/256;
% figure: mesh(CI3)
% Obținerea pragului și a măștii
level = graythresh(CI3);
BW0 = im2bw(CI3,level);
% Schimbarea domeniului imaginii inițiale
Ia = double(I)/256;
Ib = Ia.*BW0;

```

Program pentru preprocesarea și rotirea imaginilor – “preproc2”

```

% Incarcare fisier imagine
[filename,pathname] = uigetfile('*. *','Selectati fisierul imagine',300,100);
if filename ~=0
    % Citire imagine in matrice
    imagine = imread(strcat(pathname,filename));
    % Transformare din RGB in UINT8
    I = rgb2gray(imagine);
    [nr_linii, nr_coloane] = size(I);
    nr_se = 25;
    nr_rot = 10;
    filtru_gaus1;
    I1 = ordfilt2(Ib,5,ones(3,3));
    level1 = graythresh(I1);
    j2 = 1;
    for j = 1:nr_rot:180
        % Rotire imagine
        I2 = imrotate(I1,j,'bilinear','crop');
        % Transformare in bw
        BW1 = im2bw(I2,level1);
        % Eliminare gauri
        BW1a = imfill(BW1,'holes');
        qw1 = BW1a;
        qw2 = BW1a;
        qw1(floor(nr_linii/2),:) = 1;
        qw2(:,floor(nr_coloane/2)) = 1;
        qw1a = imcomplement(imfill(imcomplement(qw1),'holes'));
        qw2a = imcomplement(imfill(imcomplement(qw2),'holes'));
        BW2 = qw1a & qw2a;
        % Delimitare margini
        BW2(1:nr_se,1:nr_coloane) = 0;
        BW2(nr_linii-nr_se:nr_linii,1:nr_coloane) = 0;
        BW2(1:nr_linii,1:nr_se) = 0;
        BW2(1:nr_linii,nr_coloane-nr_se:nr_coloane) = 0;
    end
end

```

```

% Eliminarea neregularitatilor de contur ale imaginii
se = strel('disk',nr_se);
BW3 = imclose(BW2,se); bw = BW3;
% Stabilire dreptunghi de incadrare maxim posibil
j1 = 0;
for k = 1:nr_coloane - 1
    a_det = 0; b_det = 0; c_det = 0; d_det = 0;
    for l = 1:nr_linii - 1
        if a_det == 0
            if ((bw(l+1,k) - bw(l,k)) > 0)
                a_k = k+1;
                a_l = l+1;
                a_det = 1;
            end
        end
        if b_det == 0
            if ((bw(l+1,k) - bw(l,k)) < 0)
                b_k = k+1;
                b_l = l;
                b_det = 1;
            end
        end
    end
    if (a_det & b_det)
        for m = k+1:nr_coloane - 1
            if c_det == 0
                if (bw(a_l,m+1) - bw(a_l,m) < 0)
                    c_k = m;
                    c_l = a_l;
                    c_det = 1;
                end
            end
            if d_det == 0
                if (bw(b_l,m+1) - bw(b_l,m) < 0)
                    d_k = m;
                    d_l = b_l;
                    d_det = 1;
                end
            end
        end
    end
    if (a_det & b_det & c_det & d_det)
        disp('OK')
        if abs(c_k - a_k) > abs(d_k - b_k)
            lungime = abs(d_k - b_k);
            e_k = d_k;
            e_l = d_l;
        end
        if abs(c_k - a_k) <= abs(d_k - b_k)
            lungime = abs(c_k - a_k);
        end
    end
end

```

```

        e_k = c_k;
        e_l = c_l;
    end
    latime = abs(b_l - a_l);
    arie = latime*lungime;
    j1 = j1+1;
    rezultat(j1,:) = [a_k a_l lungime latime arie];
end
end
[max_arie max_linie] = max(rezultat(:,5));
a_k_max = rezultat(max_linie,1);
a_l_max = rezultat(max_linie,2);
lungime_max = rezultat(max_linie,3);
latime_max = rezultat(max_linie,4);
rezultat(:,5) = 0;
% Delimitare imagine dreptunghiulara
I3 = I2(a_l_max:a_l_max + latime_max,a_k_max:a_k_max + lungime_max);
% Eliminarea componentei de iluminare, si mentinerea componentei de reflexie
filtru_gaus;
% Salvarea imaginilor rotite intr-o structura
bd(j2).imagine = I3c;
% Normalizare Imagine
I3c = I3c - mean(mean(I3c));
% Stabilirea orientarii urmelor prelucrarii pe baza fft
proiectie = mean(I3c,1);
proiectie = proiectie(25:end-25) - mean(proiectie(25:end-25));
bd(j2).proiectie = proiectie;
s_inter = 20*log10(abs(fft(proiectie,256)).^2);
s_inter1 = s_inter;
bd(j2).s_inter = s_inter;
[s_max0(j2) f_max0(j2)] = max(s_inter1(5:floor(length(s_inter1)/2)));
j2 = j2 + 1
end
% Stabilirea unghiului de rotire al imaginii pentru
% pozitionarea urmelor prelucrarii pe directie verticala
[s_max_max j_max] = max(s_max0);
I3d = bd(j_max).imagine;
figure; imshow(I3d);
filename
% Clonare imagine
[nr_linii, nr_coloane] = size(I3d);
clona = 256;
I1cl = (I3d(1:clona,1:clona));
I2cl = (I3d(nr_linii + 1 - clona:nr_linii,1:clona));
I3cl = (I3d(1:clona,nr_coloane + 1 - clona:nr_coloane));
I4cl = (I3d(nr_linii + 1 - clona:nr_linii,nr_coloane + 1 - clona:nr_coloane));

% Salvare clone
filename1 = strcat(pathname,filename(1:length(filename)-4),'_1');
filename2 = strcat(pathname,filename(1:length(filename)-4),'_2');

```

```
filename3 = strcat(pathname.filename(1:length(filename)-4),'_3');
filename4 = strcat(pathname.filename(1:length(filename)-4),'_4');
imwrite(I1cl, strcat(filename1, '.bmp'), 'bmp');
imwrite(I2cl, strcat(filename2, '.bmp'), 'bmp');
imwrite(I3cl, strcat(filename3, '.bmp'), 'bmp');
imwrite(I4cl, strcat(filename4, '.bmp'), 'bmp');
end
```


a. Determinarea orientării prin valoarea maximă a DSP (densitatea spectrală de putere [SMITH]) calculată din media pe o direcție a valorilor de intensitate, pentru etalonul fc5x10.

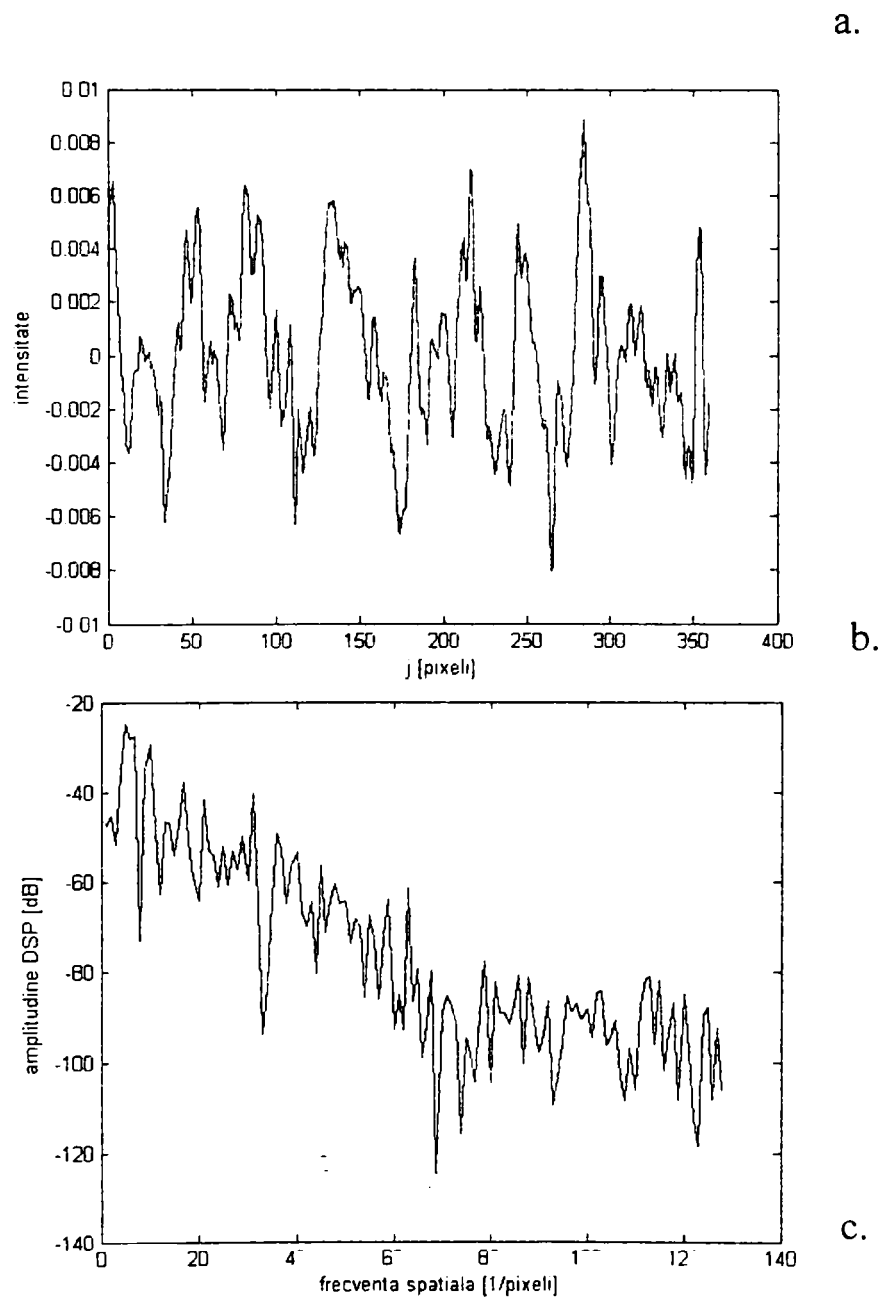


Fig.A.8.3.1. Imaginea inițială (a), media pe linii (b) și DSP a mediei (c), unghi de rotație = 0° .

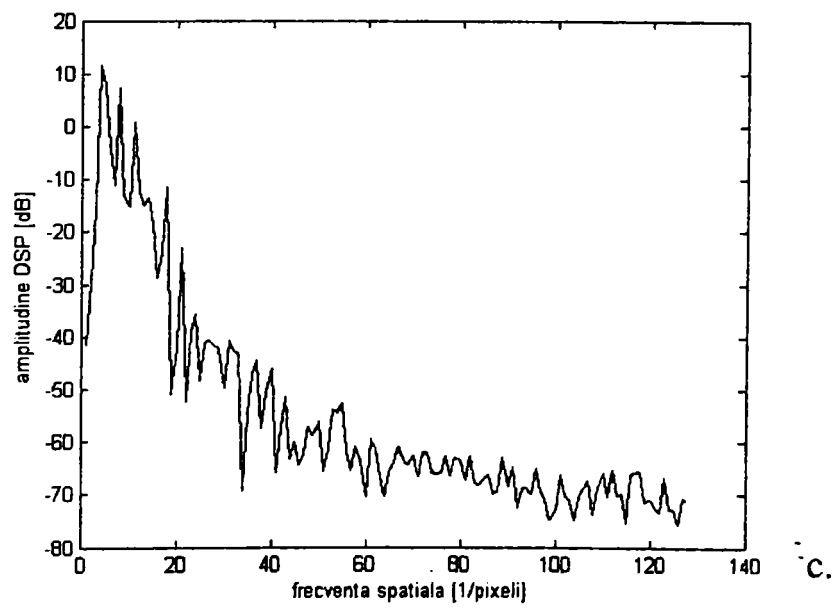
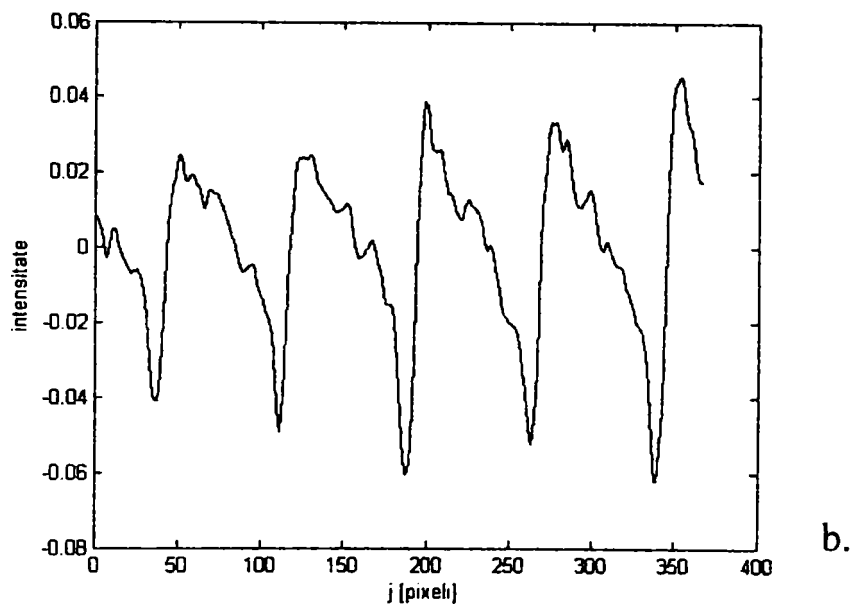
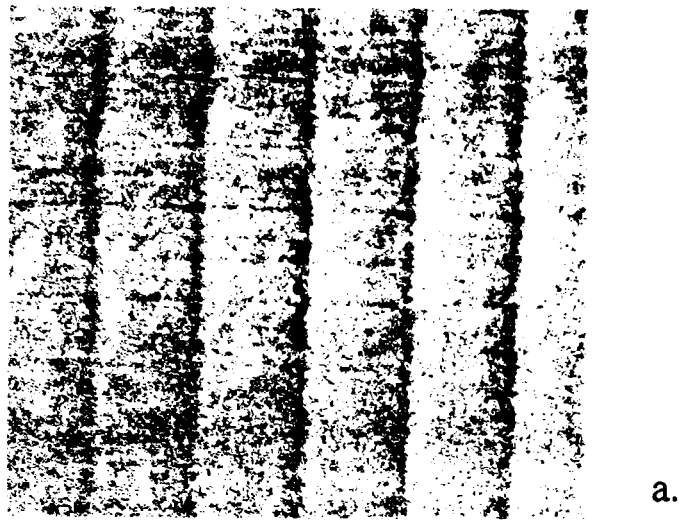
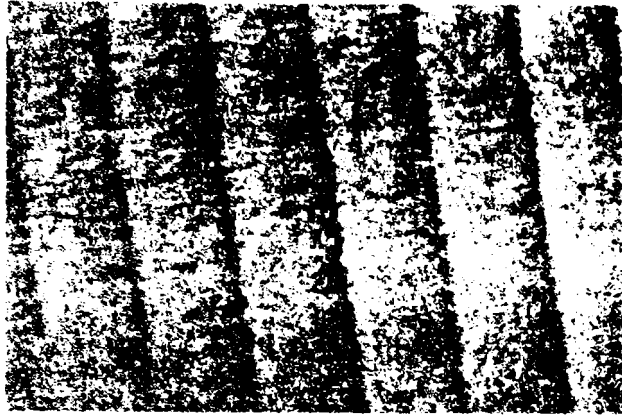
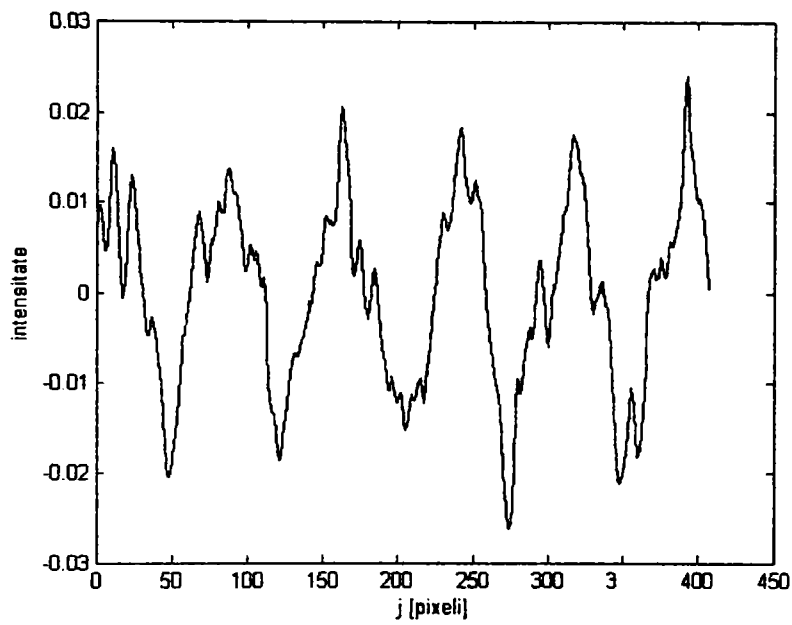


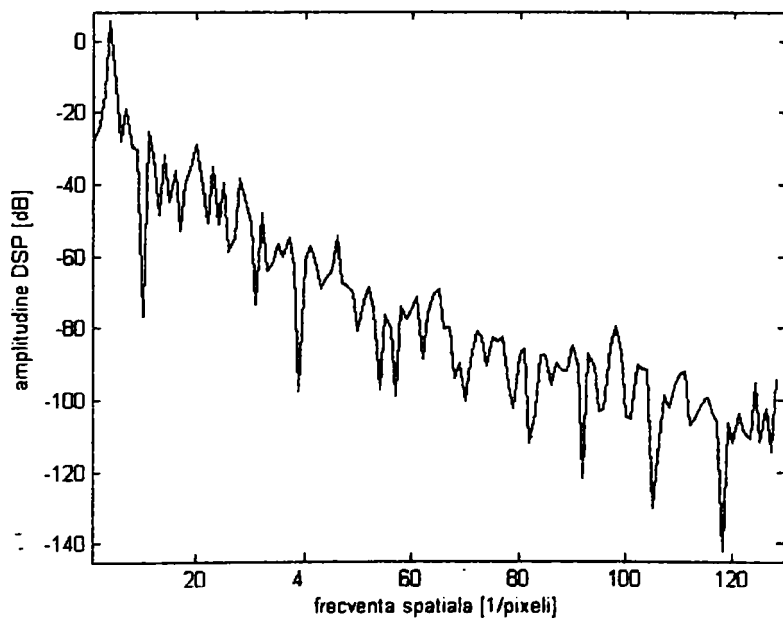
Fig.A.8.3.2. Imaginea inițială (a), media pe linii (b) și DSP a mediei (c), unghi de rotație = 10°.



a.

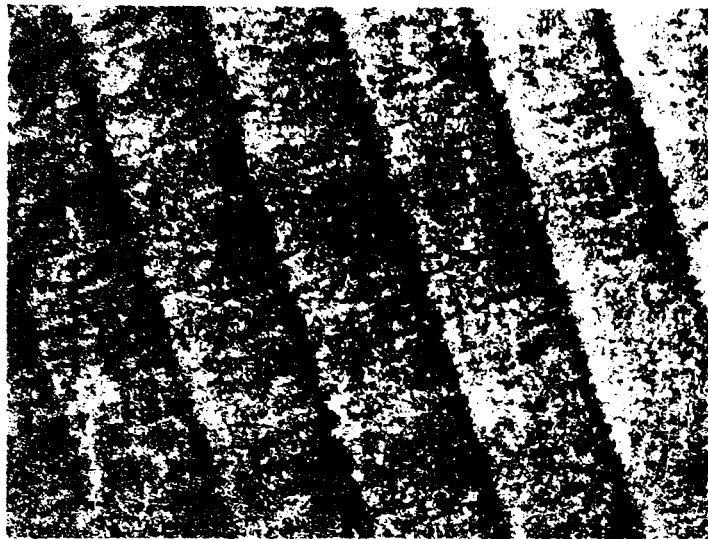


b.

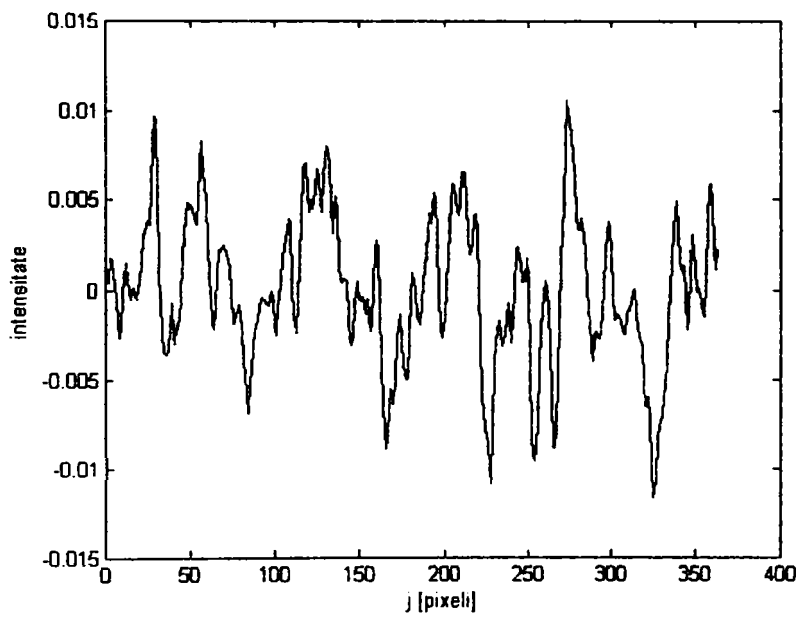


c.

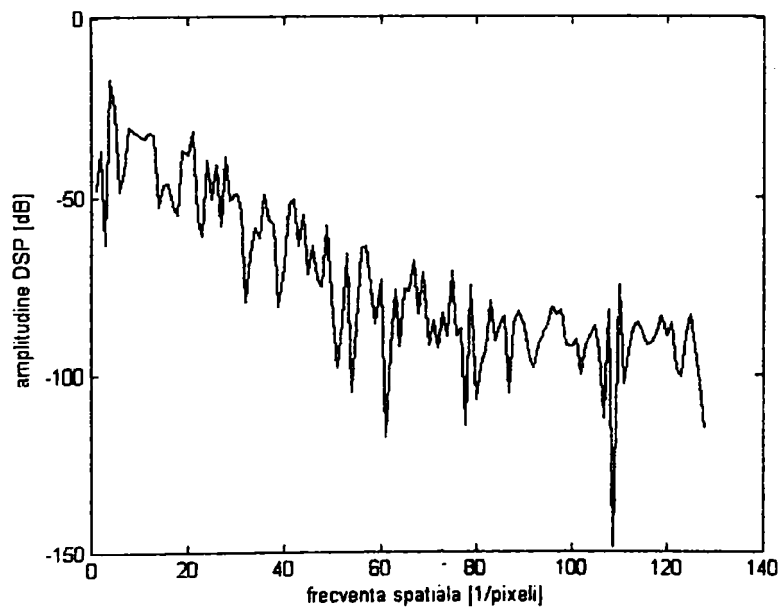
Fig.A. 8.3.3. Imaginea inițială (a), media pe linii (b) și DSP a mediei (c), unghi de rotație = 20° .



a.



b.



c.

Fig.A. 8.3.4. Imaginea inițială (a), media pe linii (b) și DSP a mediei (c), unghi de rotație = 30° .

b. Imaginile etaloanelor de rugozitate în urma preprocesării cu programul dat în ANEXA 8.2., „preproc2”.

1. Frezare cilindrică, mărire x10.

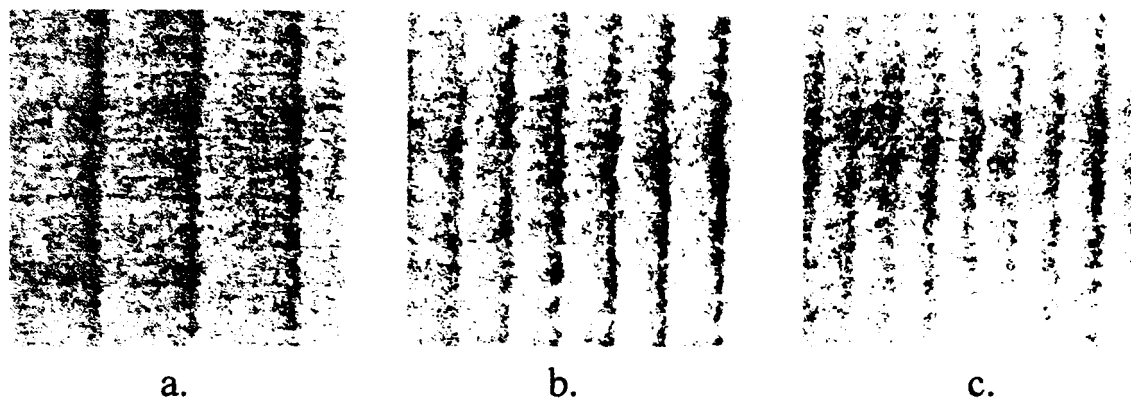


Fig.A.8.3.5. a. etalon cod : fc5x10; b. etalon cod : fc6x10; c. etalon cod : fc7x10.

2. Frezare frontală, mărire x10.

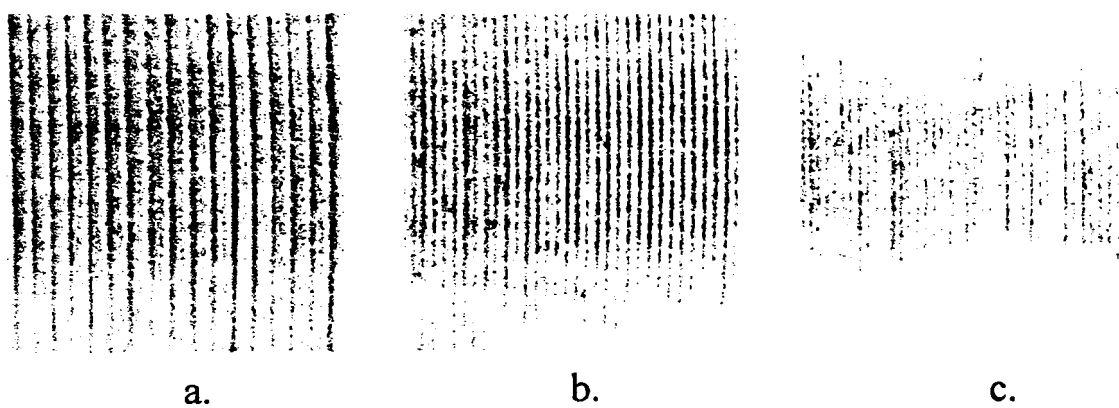


Fig.A.8.3.6. a. etalon cod : ff5x10; b. etalon cod : ff6x10; c. etalon cod : ff7x10.

3. Rabotare plană, mărire x10.

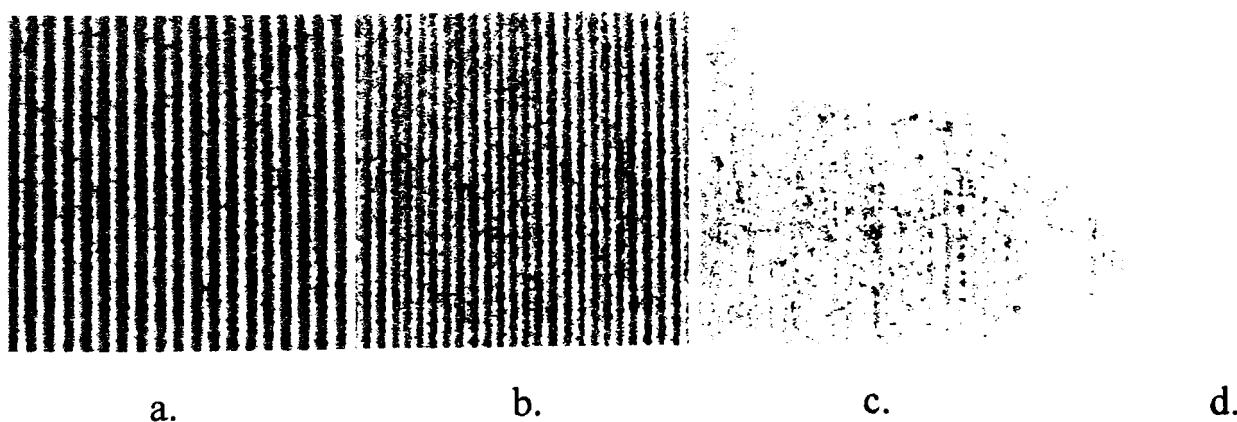
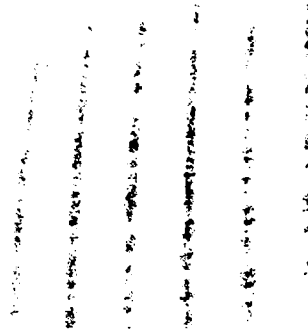


Fig.A.8.3.7. a. etalon cod : rab4x10; b. etalon cod : rab5x10; c. etalon cod : rab6x10; d. etalon cod : rab7x10.

4. Rectificare frontală, mărire x10.



a.

b.

c.

Fig.A.8.3.8. a. etalon cod : rf6x10; b. etalon cod : rf7x10; c. etalon cod : rf8x10.

5. Rectificare plană, mărire x10.

a.

b.

c.

d.

Fig.A.8.3.9. a. etalon cod : rp6x10; b. etalon cod : rp7x10; c. etalon cod : rp8x10; d. etalon cod : rp9x10.

6. Netezire plană, mărire x10.

a.

b.

c.

d.

Fig.A.8.3.10. a. etalon cod : np10x10; b. etalon cod : np11x10; c. etalon cod : np12x10; d. etalon cod : np13x10.

7. Frezare cilindrică, mărire x25.



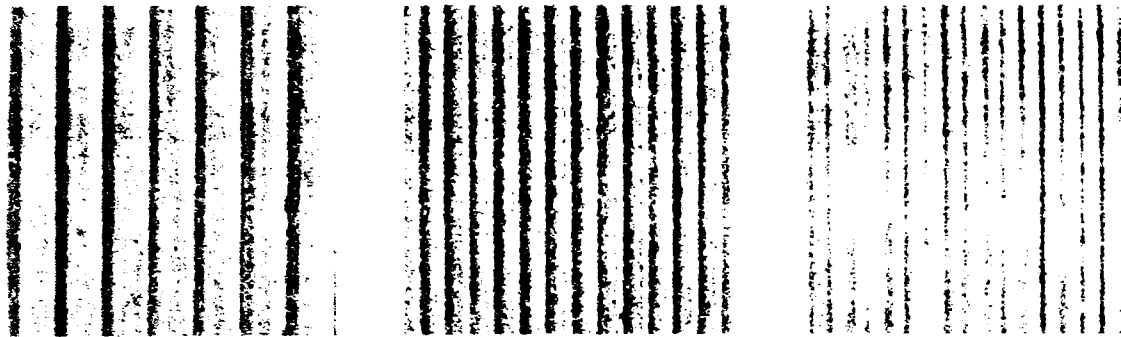
a.

b.

c.

Fig.A.8.3.11. a. etalon cod : fc5x25; b. etalon cod : fc6x25; c. etalon cod : fc7x25.

8. Frezare frontală, mărire x25.



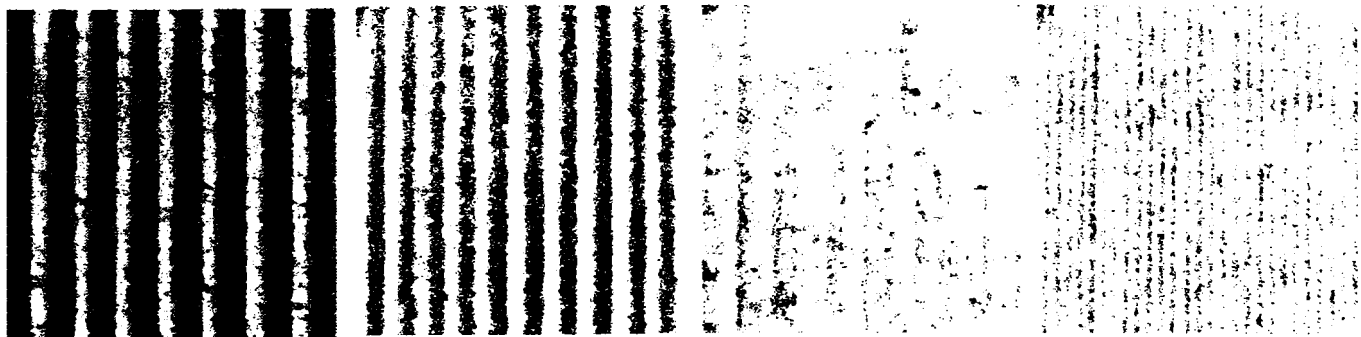
a.

b.

c.

Fig.A.8.3.12. a. etalon cod : ff5x25; b. etalon cod : ff6x25; c. etalon cod : ff7x25.

9. Rabotare plană, mărire x25.



a.

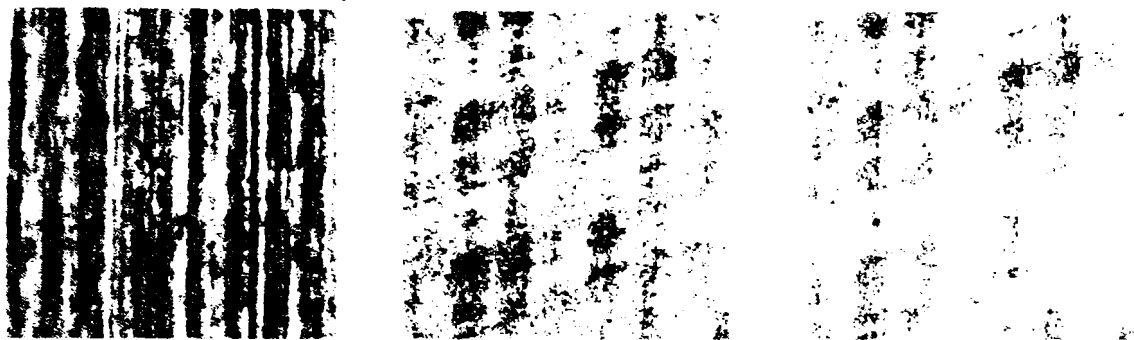
b.

c.

d.

Fig.A.8.3.13. a. etalon cod : rab4x25; b. etalon cod : rab5x25; c. etalon cod : rab6x25; d. etalon cod : rab7x25.

10. Rectificare frontală, mărire x25.



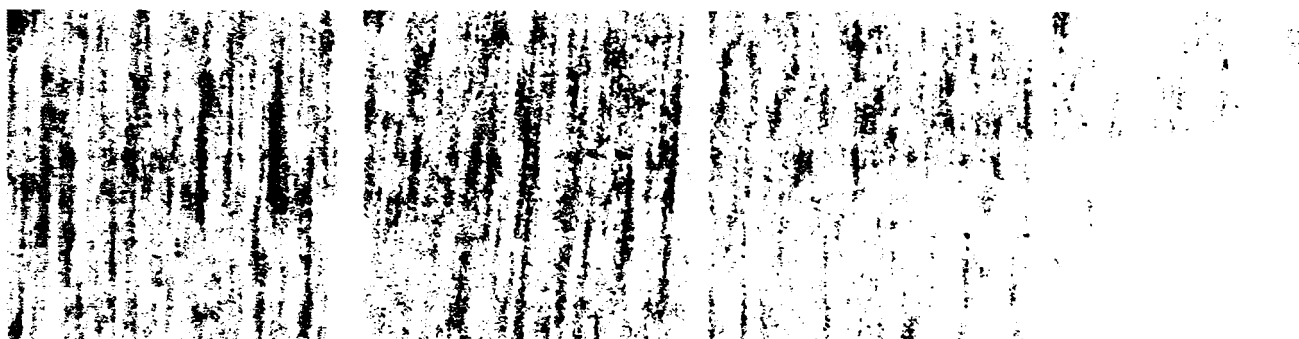
a.

b.

c.

Fig.A.8.3.14. a. etalon cod : rf6x25; b. etalon cod : rf7x25; c. etalon cod : rf8x25.

11. Rectificare plană, mărire x25.



a.

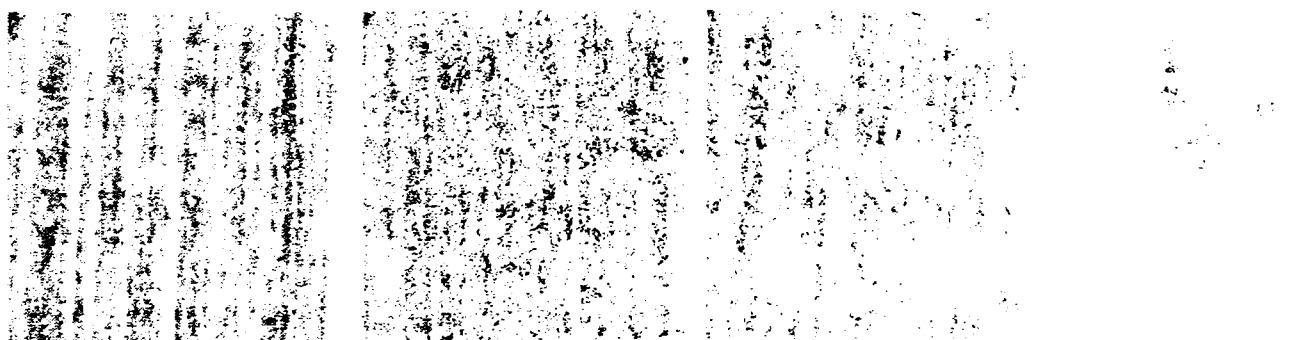
b.

c.

d.

Fig.A.8.3.15. a. etalon cod : rp6x25; b. etalon cod : rp7x25; c. etalon cod : rp8x25; d. etalon cod : rp9x25.

12. Netezire plană, mărire x25.



a.

b.

c.

d.

Fig.A.8.3.16. a. etalon cod : np10x25; b. etalon cod : np11x25; c. etalon cod : np12x25; d. etalon cod : np13x25.

Program de calcul a dimensiunii fractale prin metoda numărării casetelor.

```

% Program principal : dim_fract_KS
clear all
nnn=5; rrr=4;
global I pp
I = imread('fract5_4_3.bmp'); dim_fractal; ppp(1) = pp;
I = imread('fract5_4_4.bmp'); dim_fractal; ppp(2) = pp;
I = imread('fract5_4_5.bmp'); dim_fractal; ppp(3) = pp;
I = imread('fract5_4_6.bmp'); dim_fractal; ppp(4) = pp;
figure: plot(ppp,'k'); hold on; plot(ppp,'k*'); title('Curba lui Koch');
xlabel('Gradul de detaliere al figurii'); ylabel('Dimensiunea fractala calculata');
nnn=4; rrr=3;
I = imread('sierpinsky2f.bmp'); dim_fractal; ppp(1) = pp;
I = imread('sierpinsky2e.bmp'); dim_fractal; ppp(2) = pp;
I = imread('sierpinsky2d.bmp'); dim_fractal; ppp(3) = pp;
I = imread('sierpinsky2c.bmp'); dim_fractal; ppp(4) = pp;
I = imread('sierpinsky2b.bmp'); dim_fractal; ppp(5) = pp;
I = imread('sierpinsky2a.bmp'); dim_fractal; ppp(6) = pp;
figure: plot(ppp,'k'); hold on; plot(ppp,'k*'); title('Triunghiul lui Sierpinsky');
xlabel('Gradul de detaliere al figurii'); ylabel('Dimensiunea fractala calculata');
% Metoda de numarare a casetelor : dim_fractal
% clear all
% Initializare variabile
% indice linie
i = 1;
% indice coloana
j = 1;
% dimensiune initiala caseta
dim_cas_min = 2;
% dimensiune caseta
dim_cas = dim_cas_min;
% dimensiune finala caseta
dim_cas_max = 24;
% pasul cu care se incrementeaza dimensiunea casetei
pas_dim_cas = 1;
% numar de iteratii
nr_it = (dim_cas_max - dim_cas_min)/pas_dim_cas;
% iteratia
iteratia = 1;
% contorizare pixeli diferiti de fundal
cont_pix_on = zeros(1,nr_it);
% dimensiunea fractala
dim_fract = zeros(1,nr_it);
% se pastreaza valorile dim_cas pe masura rularii programului
dim_cas1 = zeros(1,nr_it);
% Citire imagine
% fundalul se considera negru = 0; iar obiectele albe = 1;

```

```

BW = im2bw(1.0.5);
BW1 = abs(1-double(BW));
[i_max, j_max] = size(BW1);
% h_fig = figure;
% hold on
% imshow(BW1);
% set(h_fig,'DoubleBuffer','on','Render','painters');
% h_caseta = rectangle('Position',[j,i,dim_cas,dim_cas],'EdgeColor', [1 0
0],'EraseMode','none');
% Bucla de calcul
for dim_cas = dim_cas_min: pas_dim_cas: dim_cas_max
    cont_pix_on(iteratia) = 0;
    for i = 1:dim_cas+1:i_max - dim_cas
        for j = 1:dim_cas+1:j_max - dim_cas
            bw_cas = BW1(i:i + dim_cas,j:j + dim_cas);
            if sum(sum(bw_cas)) > 1
                cont_pix_on(iteratia) = cont_pix_on(iteratia) + 1;
            % set(h_caseta, 'Position', [j,i,dim_cas,dim_cas]);
            % drawnow;
            % pause(0.01);
        end
    end
    dim_cas1(iteratia) = dim_cas;
    iteratia = iteratia + 1;
end
% Afisare rezultate
cont_pix_on2 = log(cont_pix_on);
dim_cas2 = log(1./dim_cas1);
p = polyfit(dim_cas2,cont_pix_on2,1);
f = polyval(p,dim_cas2);
pp = p(1);
pp_t = log(nnn)/log(rrr);
disp(strcat('Dim.fract. teoretica :'.num2str(pp_t)));
eroare = pp - pp_t;
disp(strcat('Eroarea :'.num2str(eroare)));
figure; plot(dim_cas2,cont_pix_on2,'k*');
hold on
plot(dim_cas2,f,'k');
title(strcat('Diagrama Richardson. Dimensiune fractala :'.num2str(pp)));
xlabel('1/r [1/pixel]');
ylabel('N');

```

Fractali utilizați pentru testarea programului “dim_fract”.

1. Curba lui Koch.

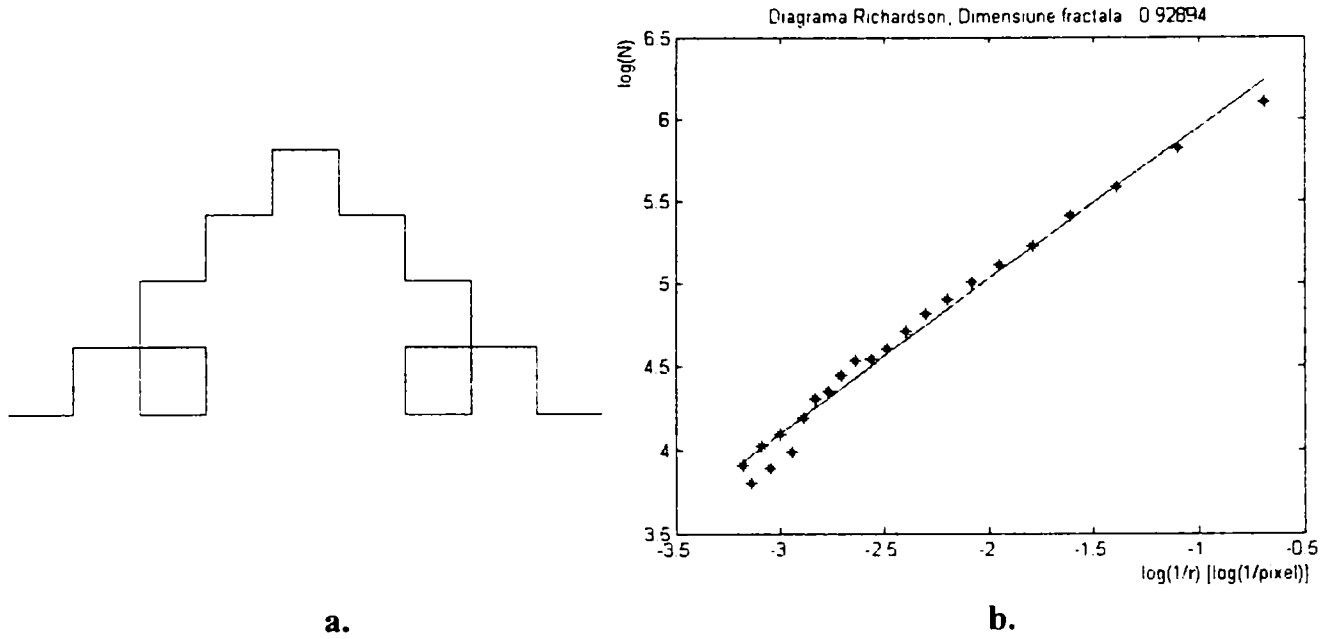


Fig.A.8.5.1. Curba lui Koch faza 1 (a); Diagrama Richardson corespunzătoare (b).

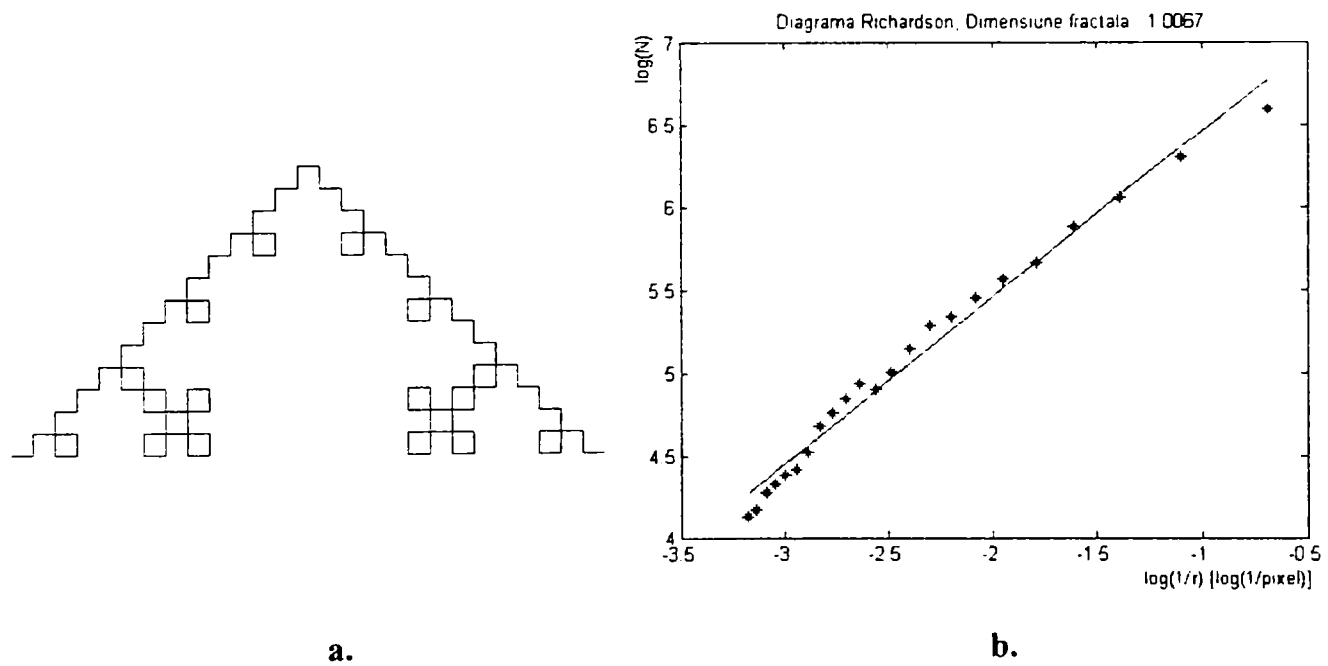
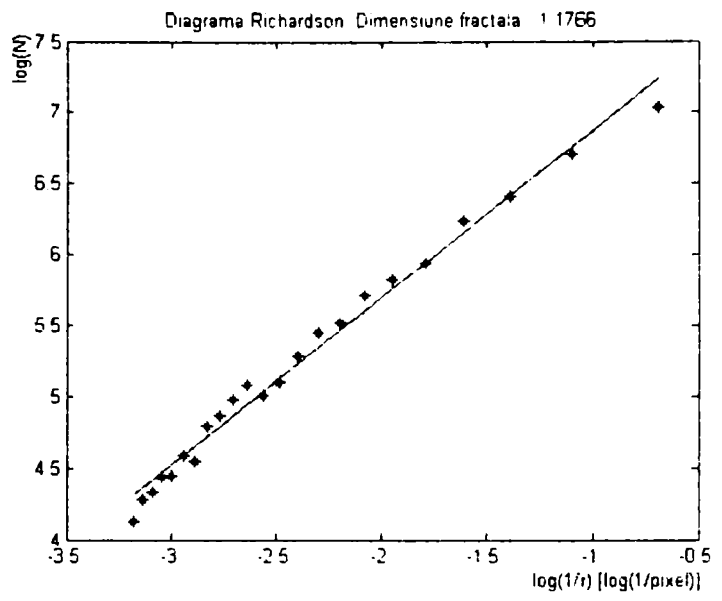
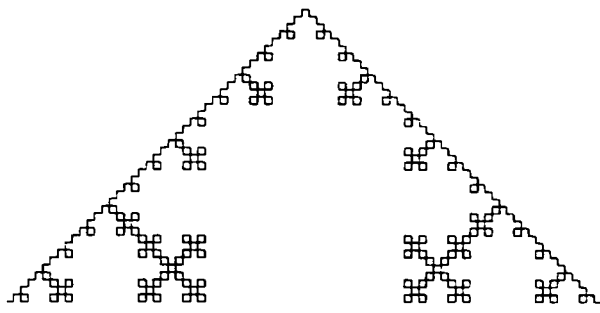


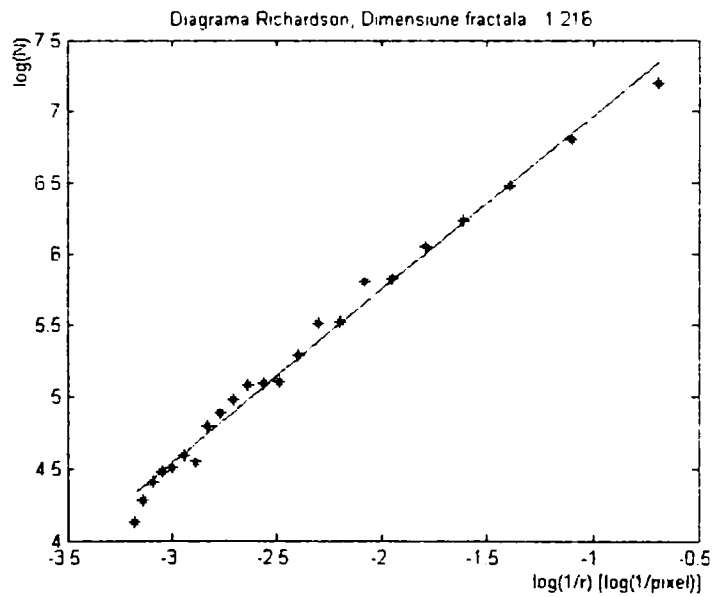
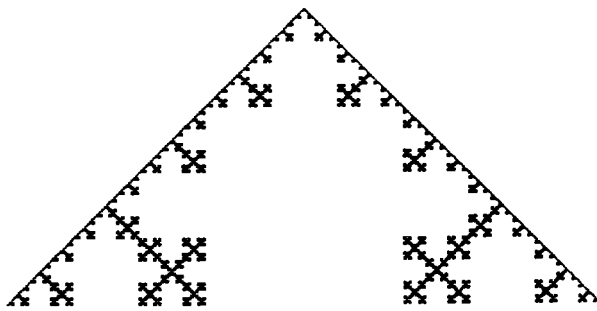
Fig.A.8.5.2. Curba lui Koch faza 2 (a); Diagrama Richardson corespunzătoare (b).



a.

b.

Fig.A.8.5.3. Curba lui Koch faza 3 (a); Diagrama Richardson corespunzătoare (b).

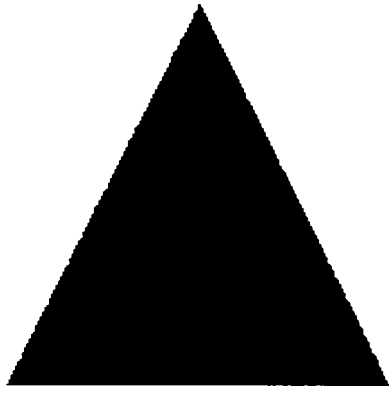


a.

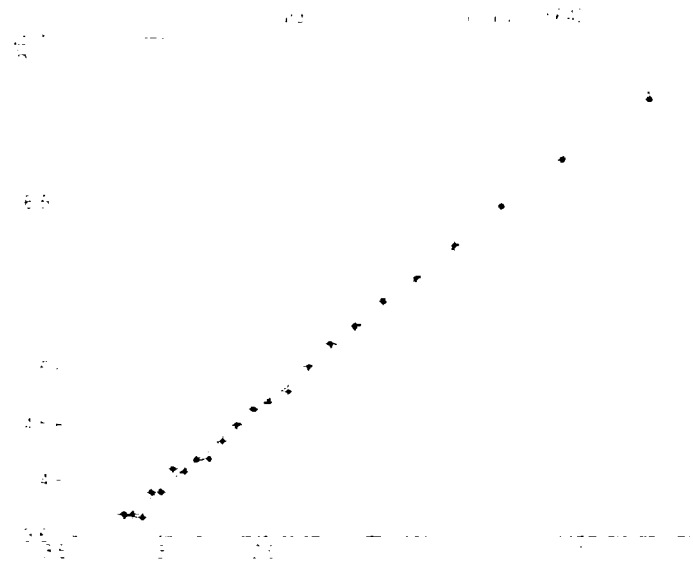
b.

Fig.A.8.5.4. Curba lui Koch faza 4 (a); Diagrama Richardson corespunzătoare (b).

2. Triunghiul lui Sierpinsky.

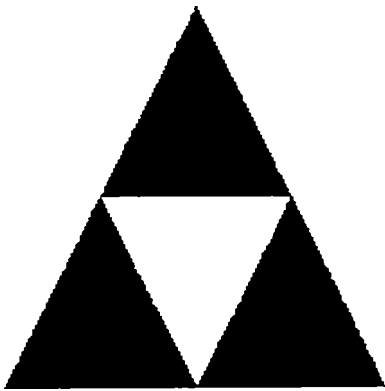


a.



b.

Fig.A.8.5.5. Triunghiul lui Sierpinsky faza 1 (a); Diagrama Richardson corespunzătoare (b).



a.

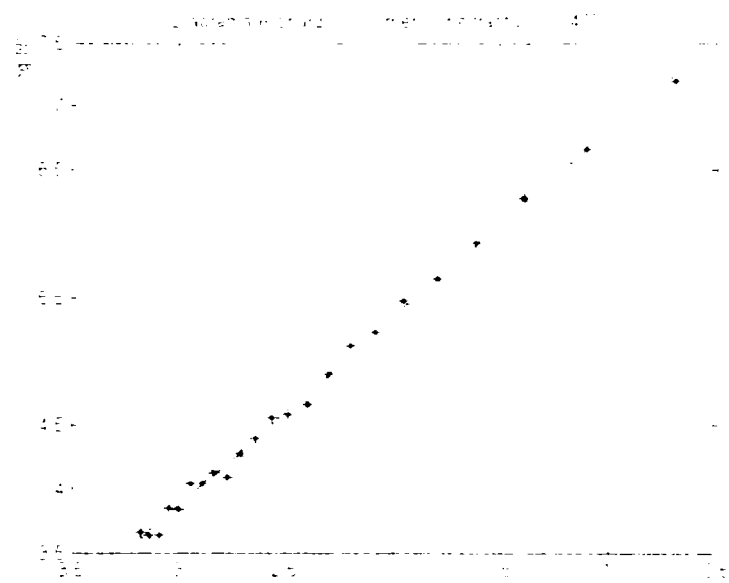
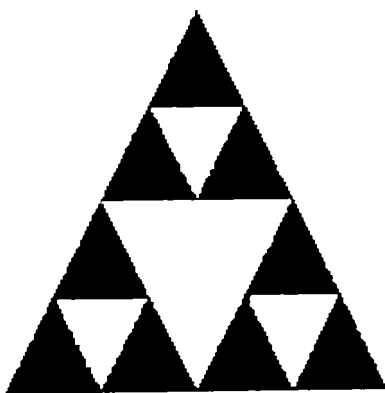


Fig.A.8.5.6. Triunghiul lui Sierpinsky faza 2 (a); Diagrama Richardson corespunzătoare (b).



a.

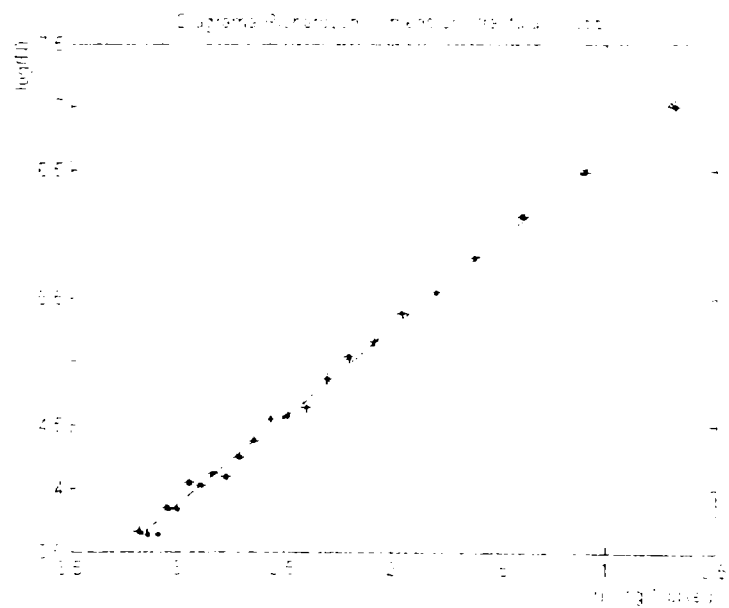
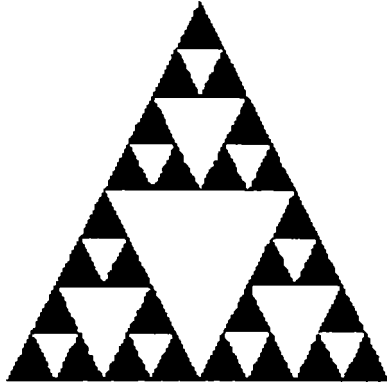
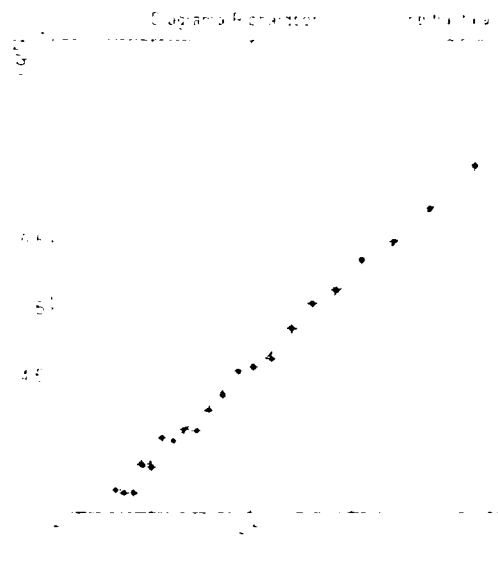


Fig.A.8.5.7. Triunghiul lui Sierpinsky faza 3 (a); Diagrama Richardson corespunzătoare (b).

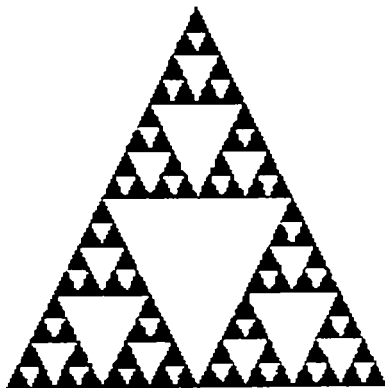


a.

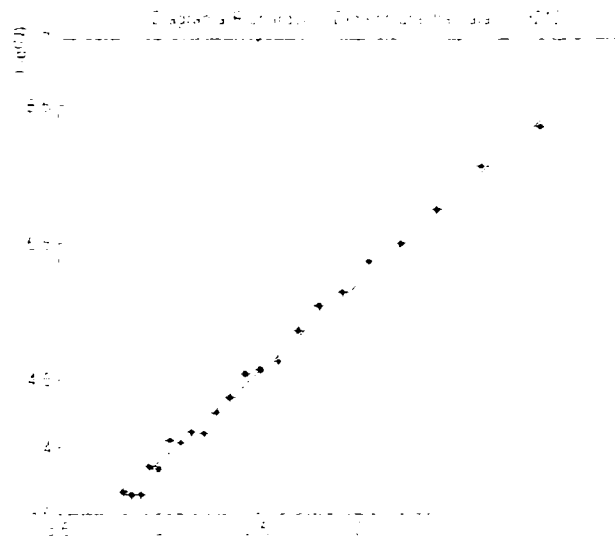


b.

Fig.A.8.5.8. Triunghiul lui Sierpinsky faza 4 (a): Diagrama Richardson corespunzătoare (b).

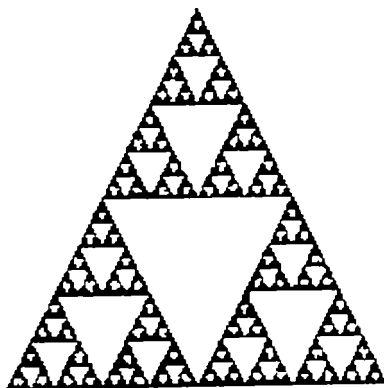


a.

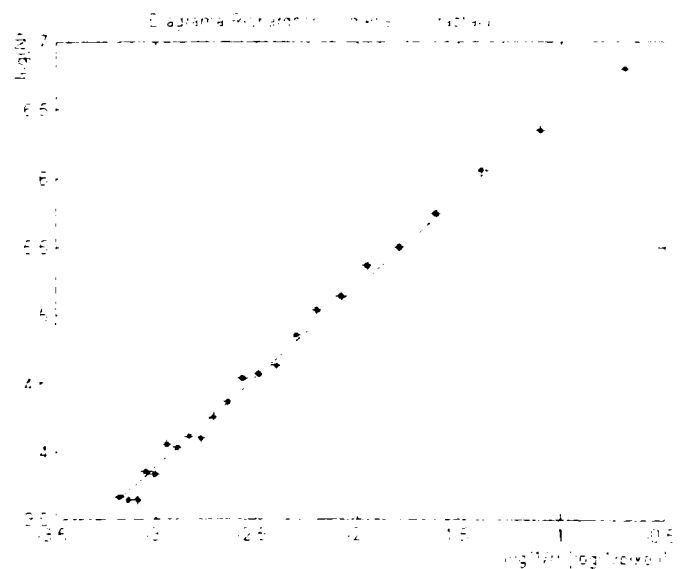


b.

Fig.A.8.5.9. Triunghiul lui Sierpinsky faza 5 (a): Diagrama Richardson corespunzătoare (b).



a.



b.

Fig.A.8.5.10. Triunghiul lui Sierpinsky faza 6 (a): Diagrama Richardson corespunzătoare (b).

Program de calcul : metoda fractală , numărarea casetelor 2D

```

addpath('C:\Labcel\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x10rotc');
global I I_a p p1
nnnn = I;
I = double(imread('fc5_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'k^'); hold on;
I = double(imread('fc6_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'kv');
I = double(imread('fc7_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'ks');
I = double(imread('ff5_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'b^');
I = double(imread('ff6_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'bv');
I = double(imread('ff7_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'bs');
I = double(imread('rab4_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'cd');
I = double(imread('rab5_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'c^');
I = double(imread('rab6_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'cv');
I = double(imread('rab7_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'cs');
I = double(imread('rf6_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'mv');
I = double(imread('rf7_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'ms');
I = double(imread('rf8_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'mo');
I = double(imread('rp6_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'rv');
I = double(imread('rp7_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'rs');
I = double(imread('rp8_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'ro');
I = double(imread('rp9_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'r*');
I = double(imread('np10_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'g+');
I = double(imread('np11_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'gx');
I = double(imread('np12_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'g.');
I = double(imread('np13_x10.bmp')); nc1D1;
plot(p1(1),p2(1),'g<');

```

Subprogram nc1D1

```
I_a = ones(256,256);
clear p10 p20
k = 1;
pas = 2;
for n = pas:pas:256
    lim1 = n - pas + 1;
    lim2 = n;
    I1 = floor(mean(I(lim1:lim2,:),1));
    for m = 2:256
        if I1(m) >= I1(m-1)
            I_a(m,I1(m-1):I1(m)) = 0;
        else
            I_a(m,I1(m):I1(m-1)) = 0;
        end
    end
end
nc1D2:
p10(k,:) = p(1,1);
p20(k,:) = p(1,2);
k = k + 1;
I_a = ones(256,256);
end
p1 = mean(p10,1)
p2 = mean(p20,1)
```


Subprogram nc1D2

```
% Metoda de numarare a casetelor
% Initializare variabile
% indice linie
i = 1;
% indice coloana
j = 1;
% dimensiune initiala caseta
dim_cas_min = 1;
% dimensiune caseta
dim_cas = dim_cas_min;
% dimensiune finala caseta
dim_cas_max = 24;
% pasul cu care se incrementeaza dimensiunea casetei
pas_dim_cas = 1;
% numar de iteratii
nr_it = (dim_cas_max - dim_cas_min)/pas_dim_cas;
% iteratia
iteratia = 1;
% contorizare pixeli diferiti de fundal
cont_pix_on = zeros(1,nr_it);
% dimensiunea fractala
dim_fract = zeros(1,nr_it);
% se pastreaza valorile dim_cas pe masura rularii programului
dim_cas1 = zeros(1,nr_it);
% Citire imagine
% fundalul se considera negru = 0; iar obiectele albe = 1;
% I = imread('fract5_4_6.bmp');
BW = im2bw(I_a,0.5);
%imshow(BW);
BW1 = abs(1-double(BW));
[i_max, j_max] = size(BW1);
% h_fig = figure;
% hold on
% imshow(imcomplement(BW1));
% set(h_fig,'DoubleBuffer','on','Render','painters');
% h_caseta = rectangle('Position',[j,i,dim_cas,dim_cas],'EdgeColor',[1 0
0],'EraseMode','none');
% Bucla de calcul
for dim_cas = dim_cas_min: pas_dim_cas: dim_cas_max
    cont_pix_on(iteratia) = 0;
    for i = 1:dim_cas+1:i_max - dim_cas
        for j = 1:dim_cas+1:j_max - dim_cas
            bw_cas = BW1(i:i + dim_cas,j:j + dim_cas);
            if sum(sum(bw_cas)) > 1
                cont_pix_on(iteratia) = cont_pix_on(iteratia) + 1;
                set(h_caseta, 'Position', [j,i,dim_cas,dim_cas]);
            % drawnow;
            % pause(0.1);
        end
    end
end
```

```
        end
    end
end
dim_cas1(iteratia) = dim_cas:
iteratia = iteratia + 1:
end
```

```
% Afișare rezultate
```

```
cont_pix_on2 = log10(cont_pix_on);
dim_cas2     = log10(1./dim_cas1);
% figure:
% plot(dim_cas2,cont_pix_on2,'k*');
% hold on
p = polyfit(dim_cas2,cont_pix_on2,1);
f = polyval(p,dim_cas2);
% eroare = f - cont_pix_on2;
% plot(dim_cas2,f,'k');
% title('Diagrama Richardson')
% xlabel('1/r [1/pixeli]')
% ylabel('N');
% figure: plot(eroare,'k*');
```

Diagrame Richardson pentru imaginea fc5x10 (linia i = 64), în funcție de diferite domenii de variație a dimensiunii casei.

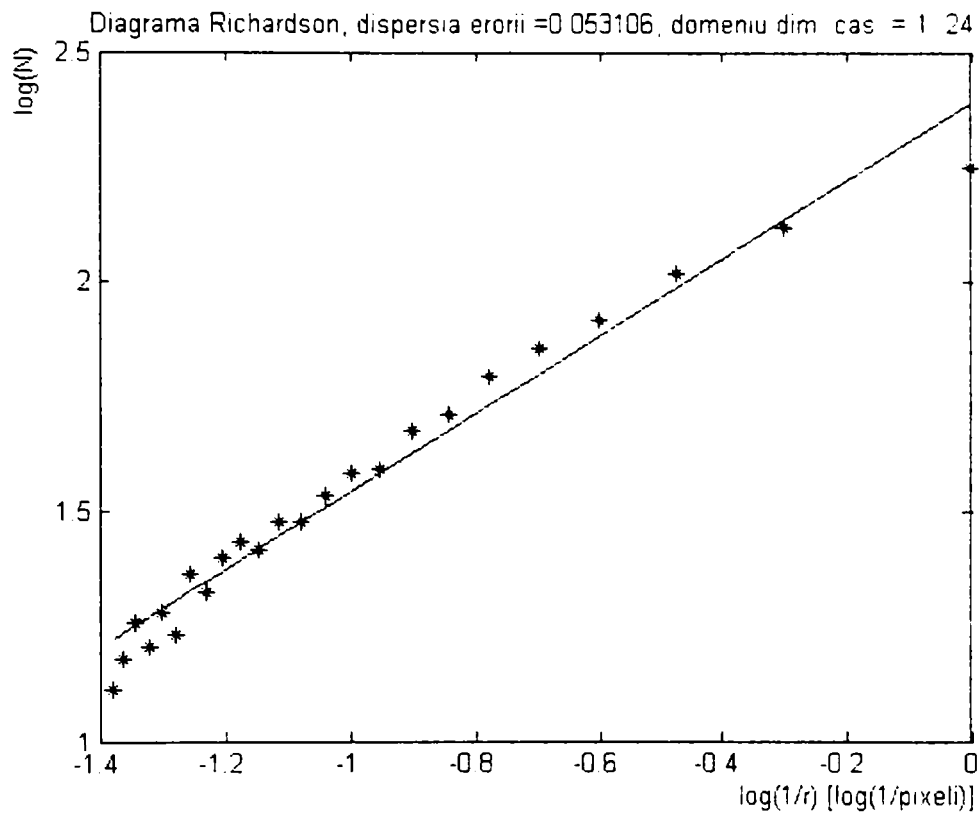


Fig.A.8.7.1. Diagrama Richardson : domeniu de variație a dim. casei = 1..24 pixeli, dispersia erorii = 0.053106.

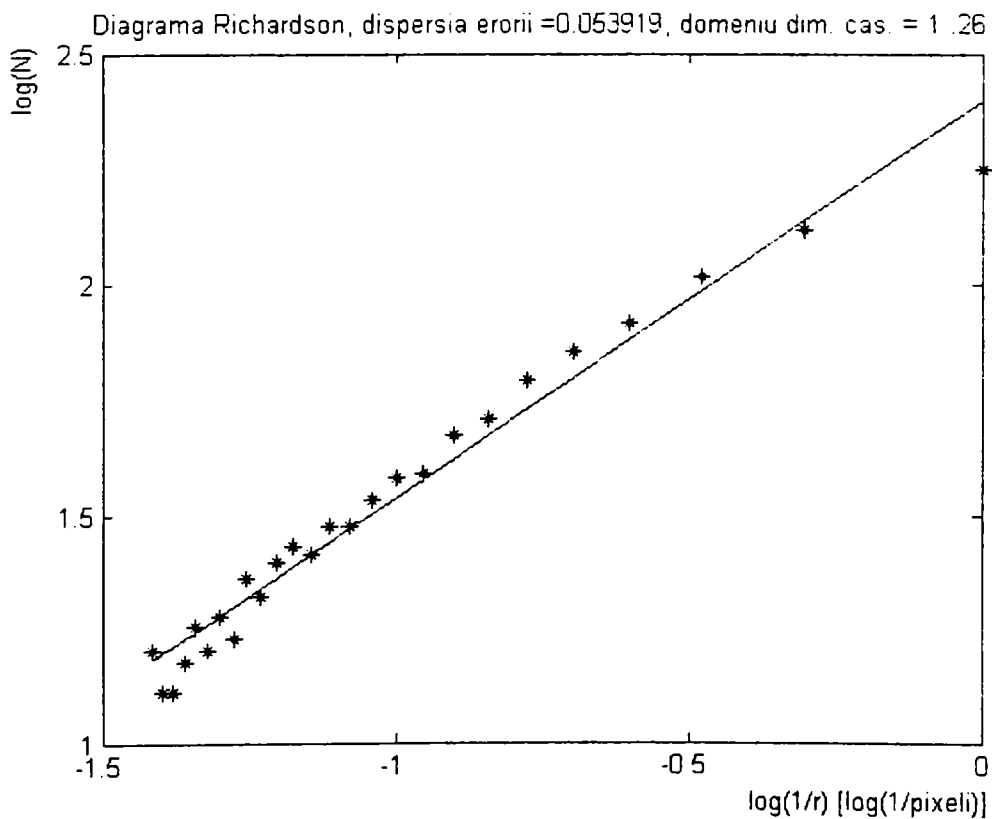


Fig.A.8.7.2. Diagrama Richardson : domeniu de variație a dim. casei = 1..26 pixeli, dispersia erorii = 0.053919

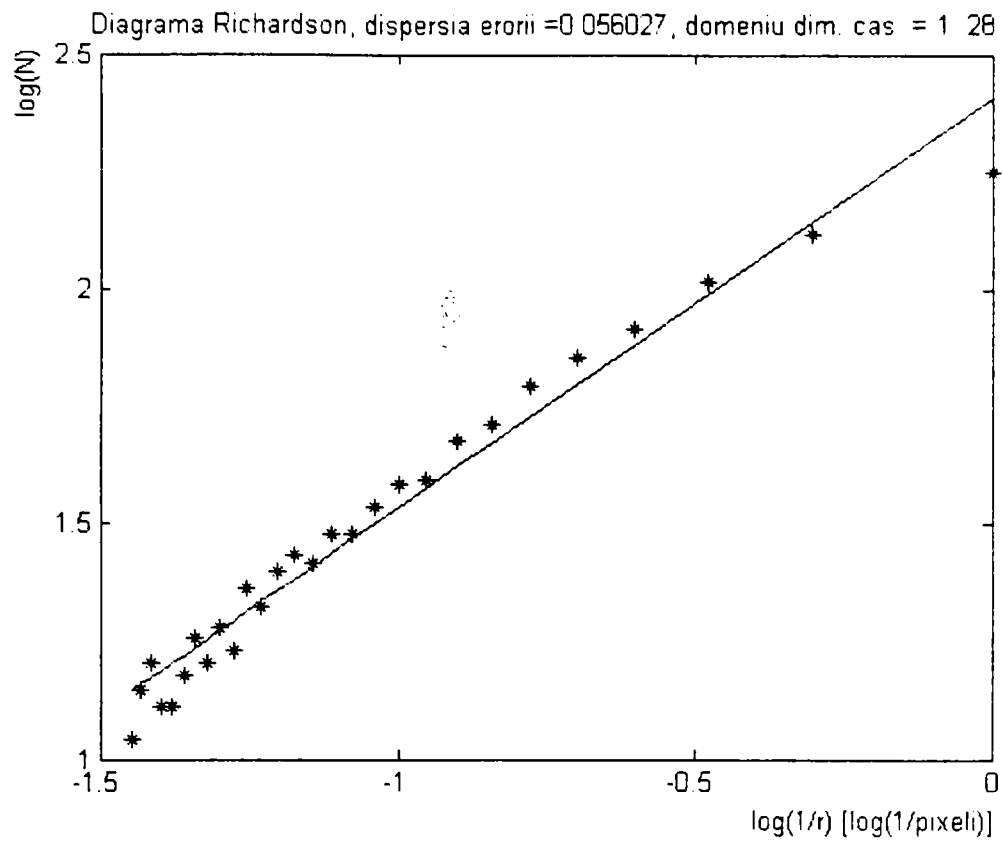


Fig.A.8.7.3. Diagrama Richardson : domeniu de variație a dim. casetei = 1..28 pixeli.
dispersia erorii =0.056027

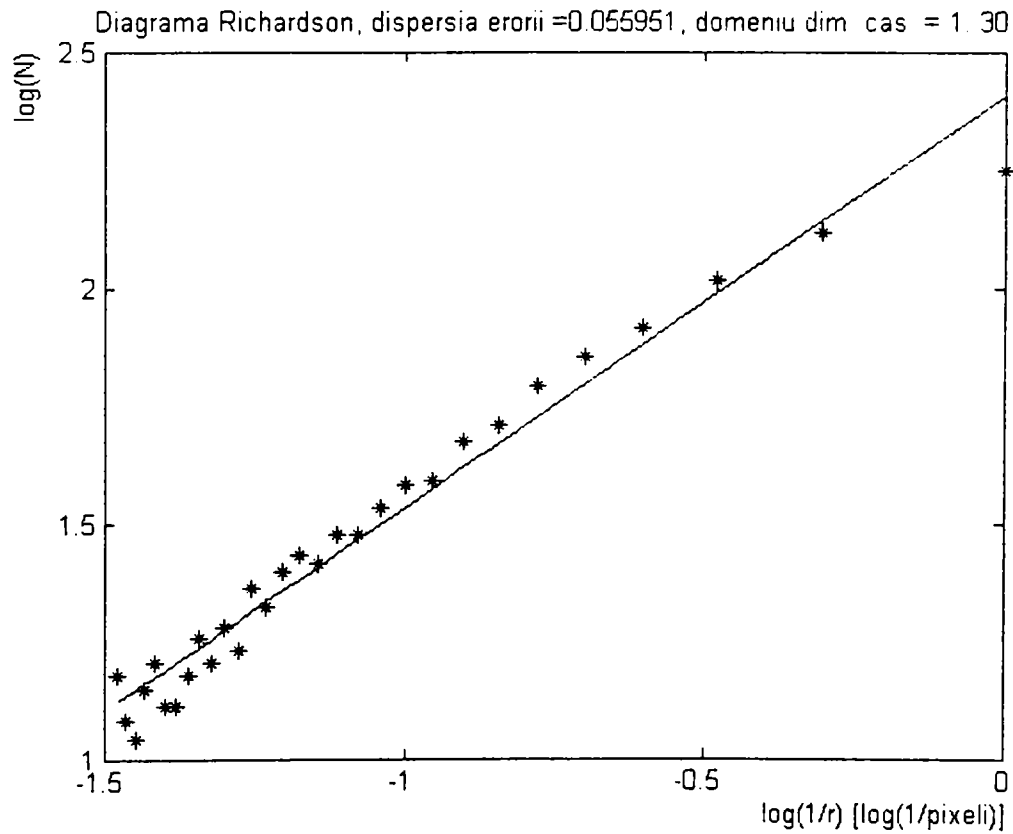


Fig.A.8.7.4. Diagrama Richardson : domeniu de variație a dim. casetei = 1..30 pixeli,
dispersia erorii =0,055951

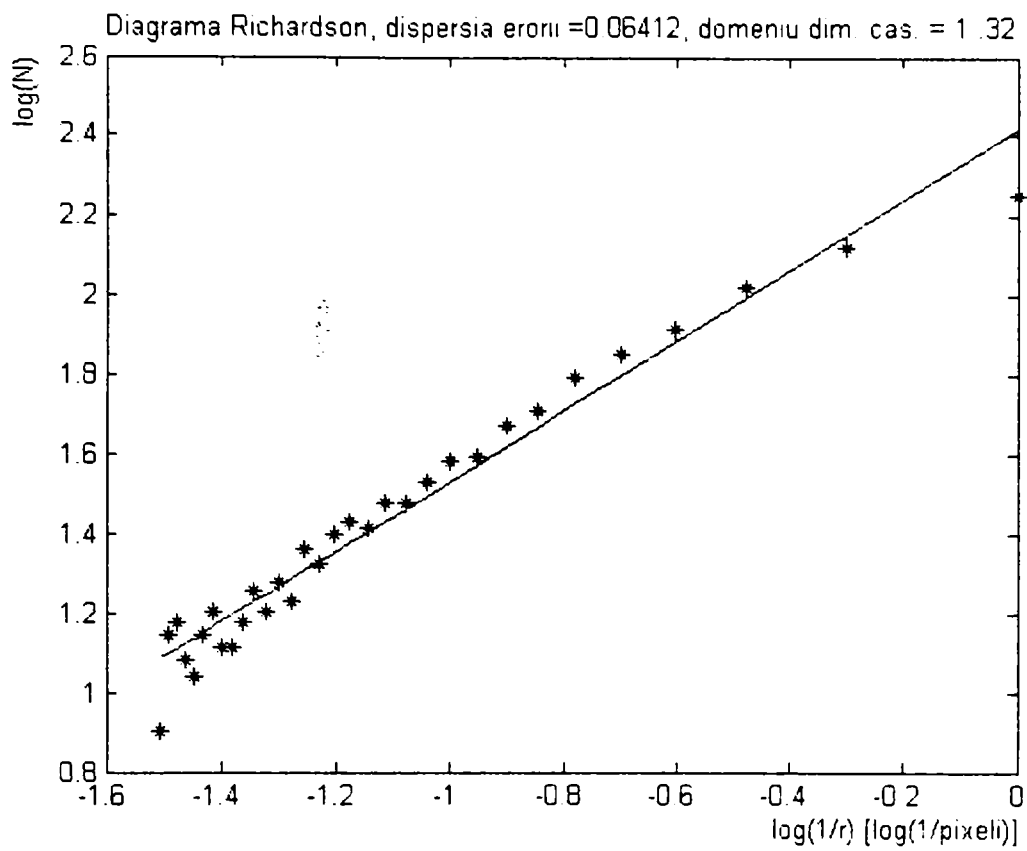


Fig.A.8.7.5. Diagrama Richardson : domeniu de variație a dim. casetei = 1.32 pixeli, dispersia erorii =.0,06412

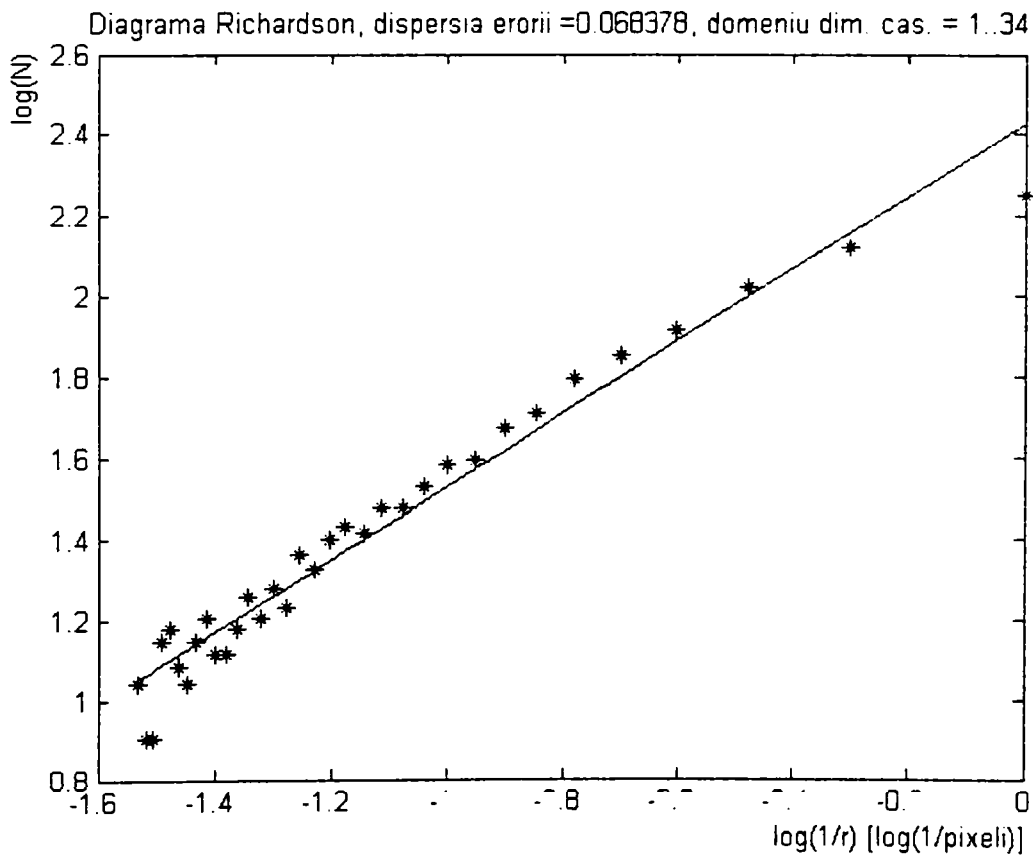


Fig.A.8.7.6. Diagrama Richardson : domeniu de variație a dim. casetei = 1.34 pixeli, dispersia erorii =.0,068378.

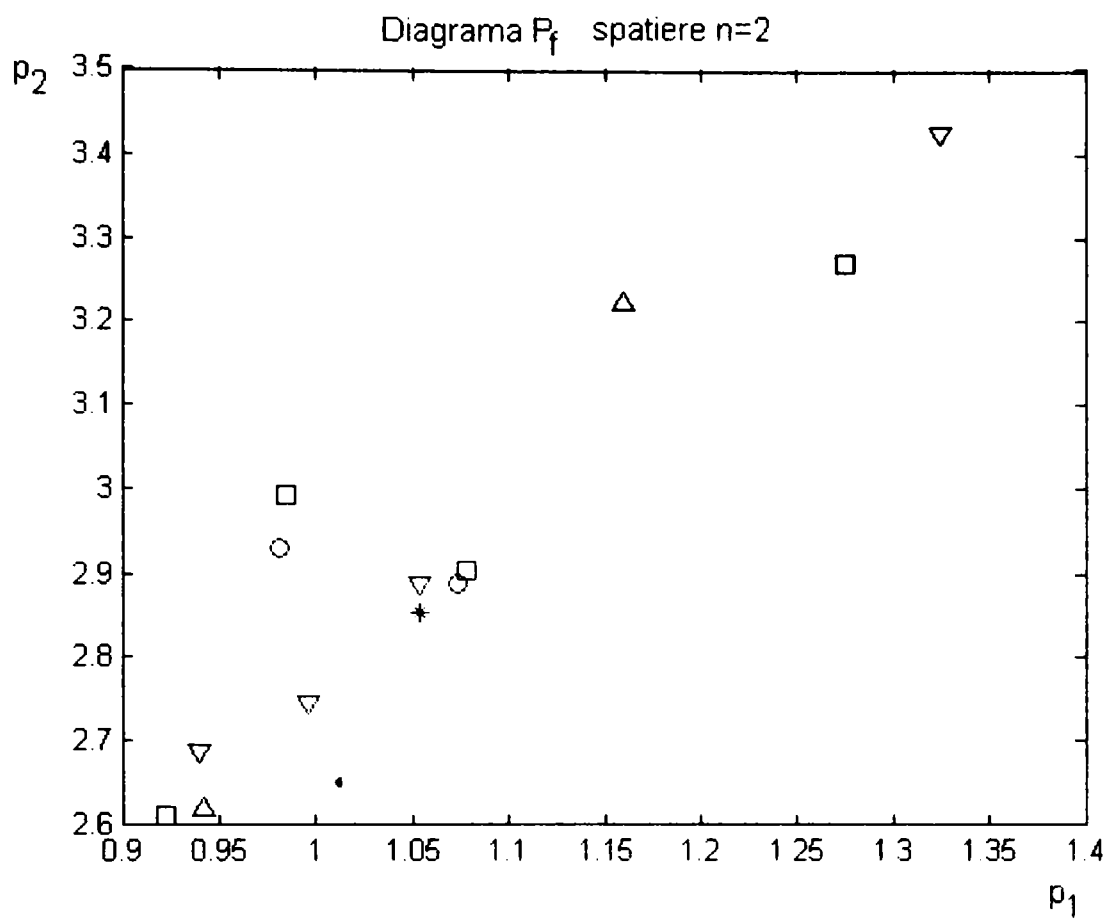


Fig.A.8.7.7. Diagrama P_f , pentru etaloanele cu marire de $\times 10$, spațiere $n = 2$.

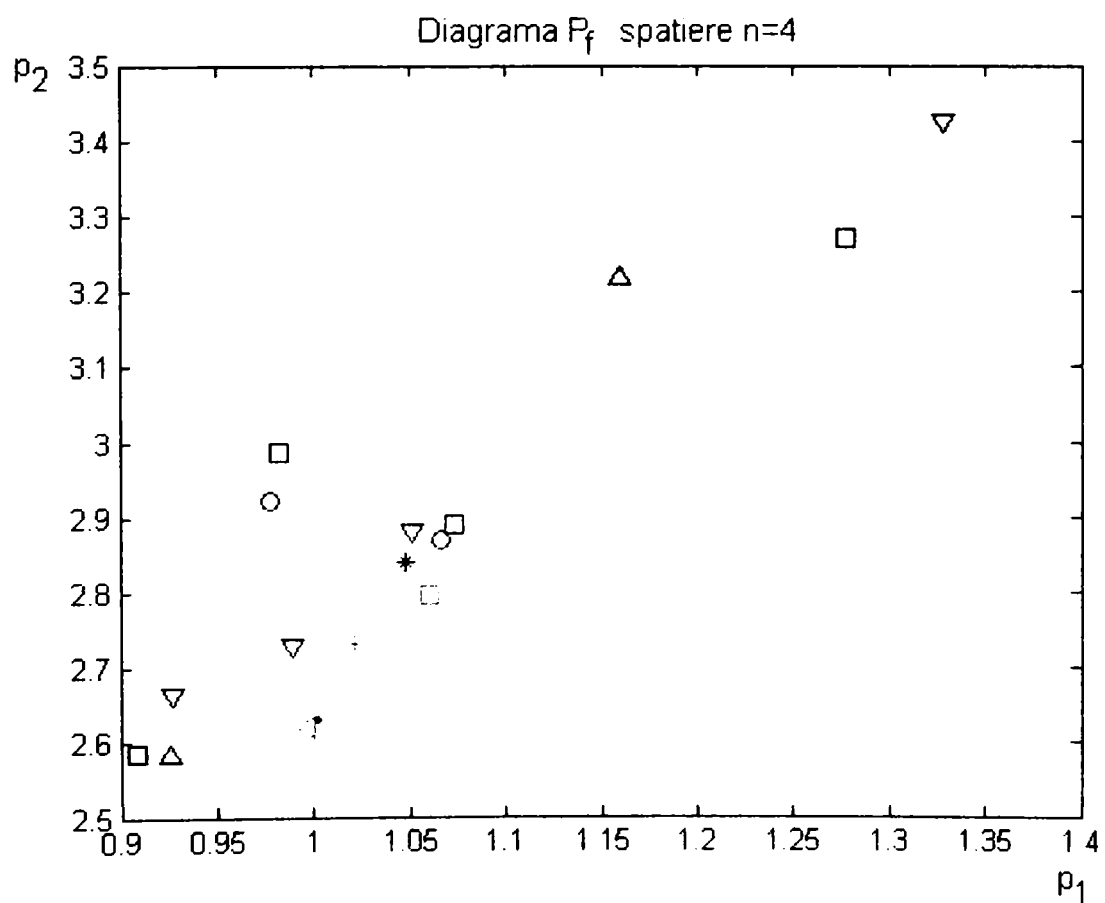


Fig.A.8.7.8. Diagrama P_f , pentru etaloanele cu marire de $\times 10$, spațiere $n = 4$.

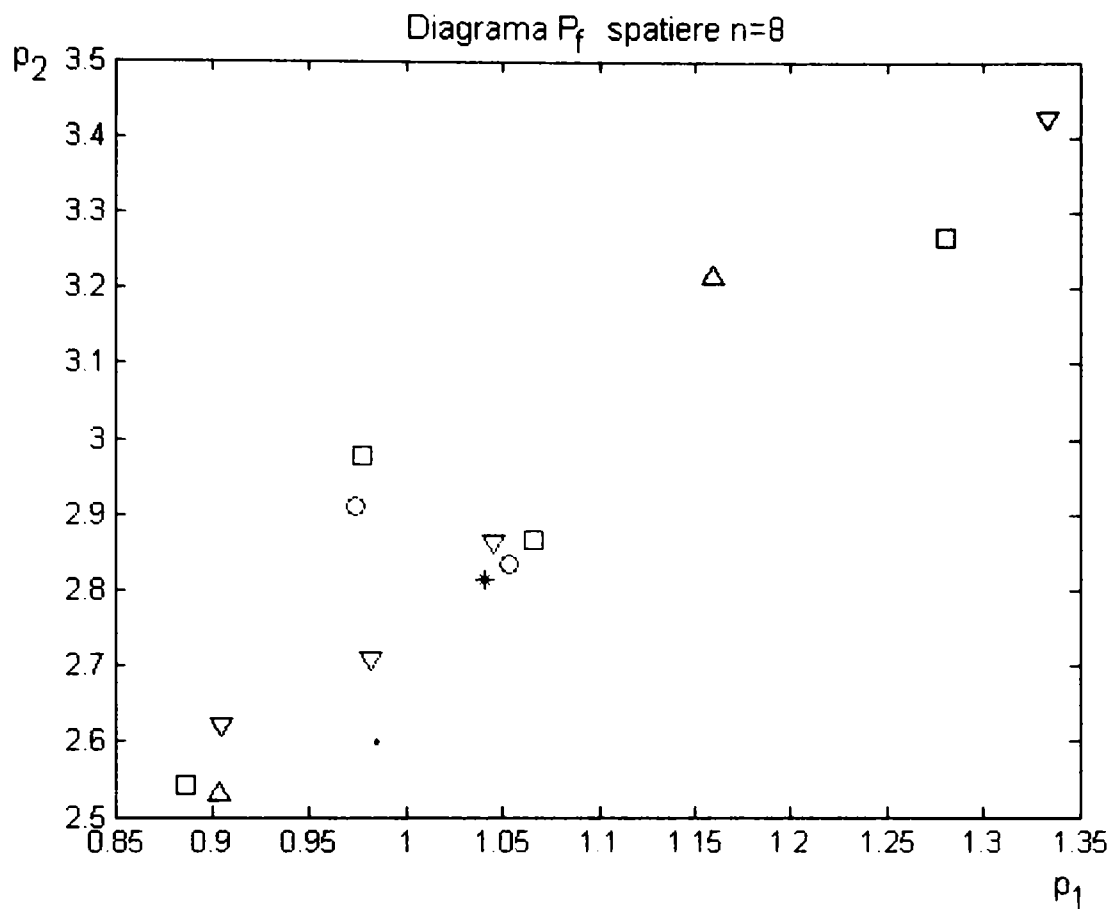


Fig.A.8.7.9. Diagrama P_f , pentru etaloanele cu marire de $\times 10$, spatiere $n = 8$.

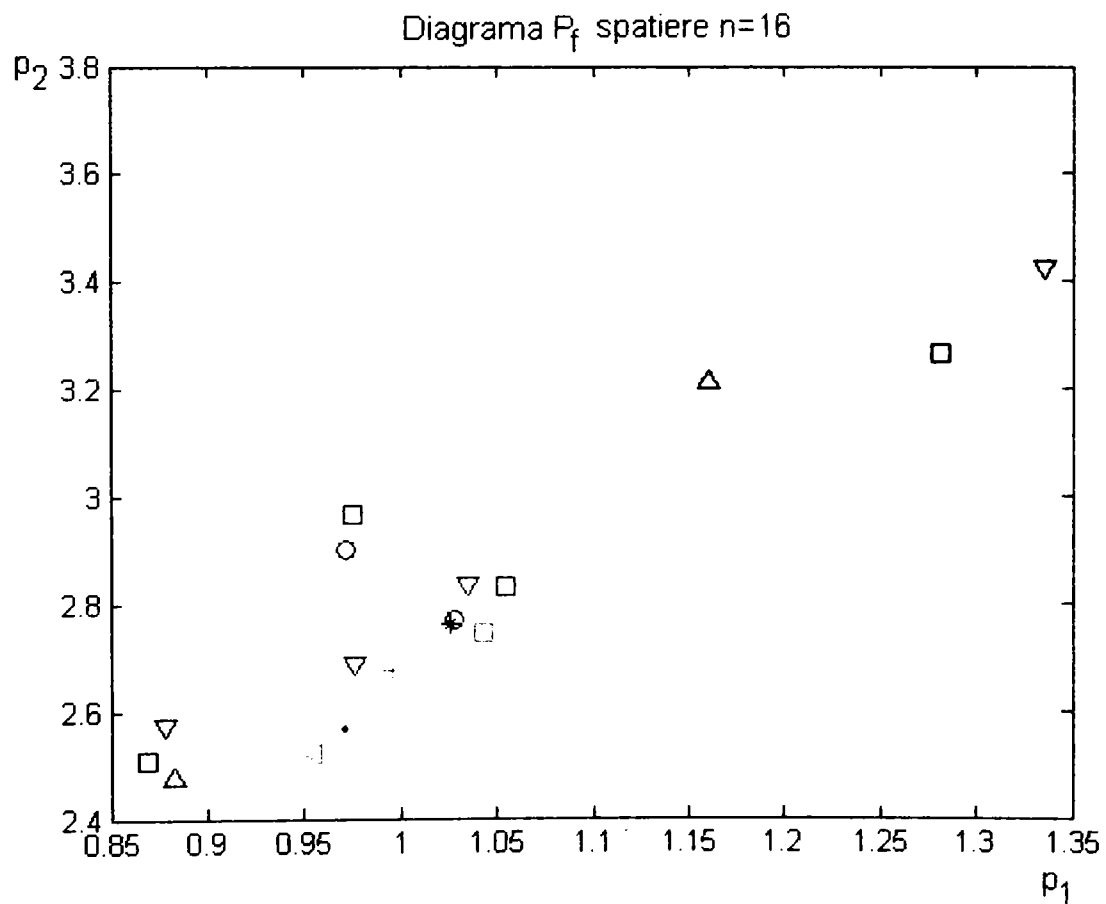


Fig.A.8.7.10. Diagrama P_f , pentru etaloanele cu marire de $\times 10$, spatiere $n = 16$.

Reprezentări ale masivului M_b obținute pentru etalonul fc5x10

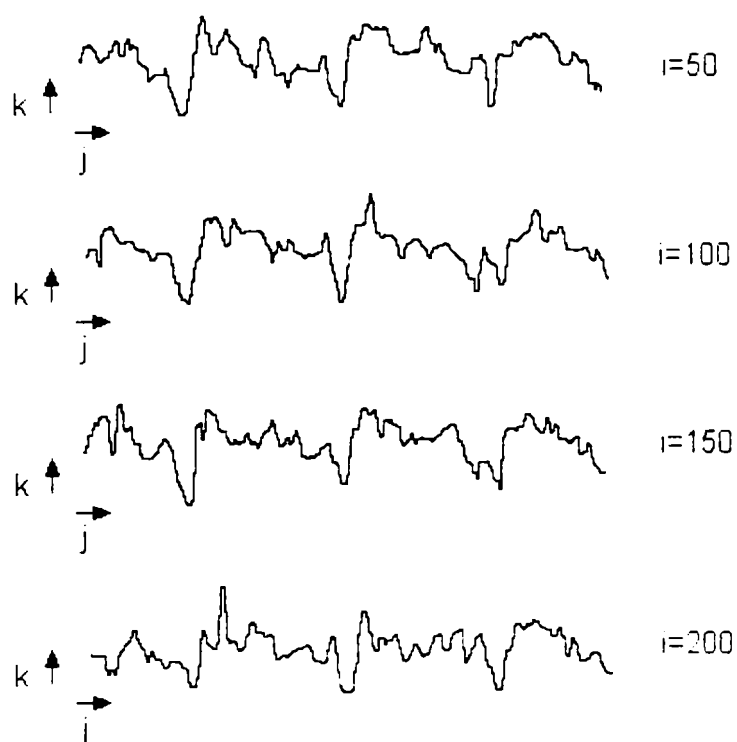


Fig.A.8.8.1. Diferite secțiuni prin masiv pentru i fixat.

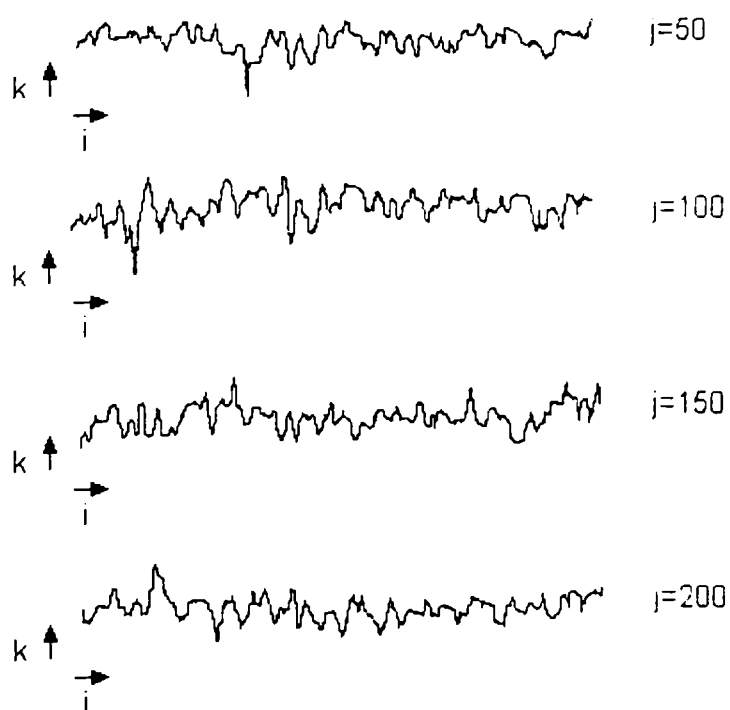


Fig.A.8.8.2. Diferite secțiuni prin masiv pentru j fixat.

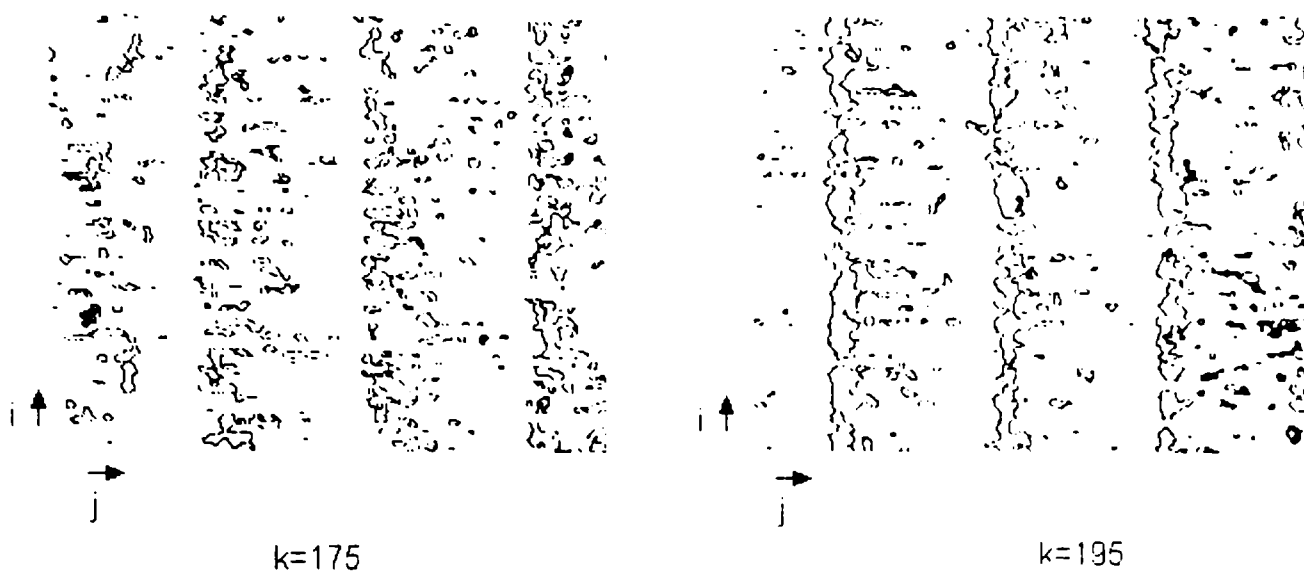


Fig.A.8.8.3. Diferite secțiuni prin masiv pentru k fixat.

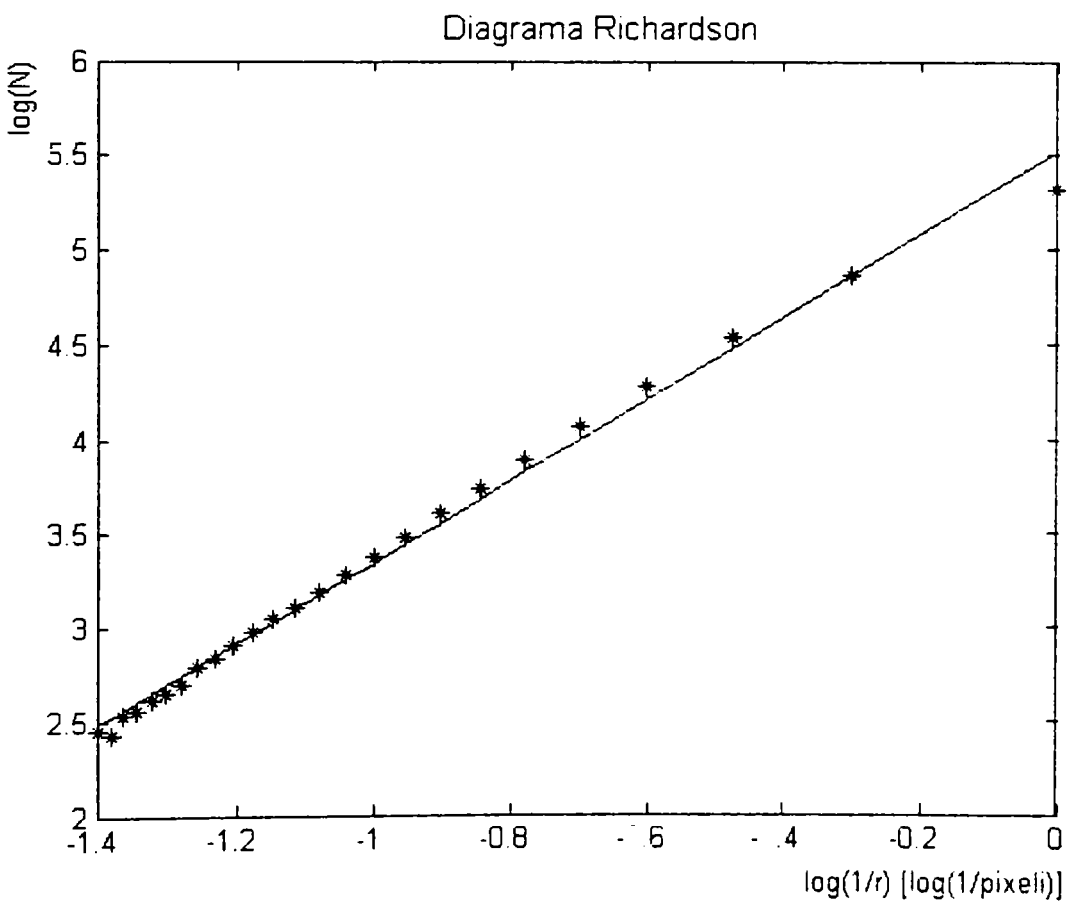


Fig.A.8.8.4. Diagrama Richardson calculată cu programul MNC3D pentru etalonul fc5x10.

Program MNC3D

```

addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x10rotd');
% addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x25rotd');
global I I_a p pl
I = double(imread('fc5_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'k^'); hold on;
I = double(imread('fc6_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'kv'); hold on;
I = double(imread('fc7_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'ks');
I = double(imread('ff5_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'b^');
I = double(imread('ff6_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'bv');
I = double(imread('ff7_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'bs');
I = double(imread('rab4_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'cd');
I = double(imread('rab5_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'c^');
I = double(imread('rab6_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'cv');
I = double(imread('rab7_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'cs');
I = double(imread('rf6_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'mv');
I = double(imread('rf7_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'ms');
I = double(imread('rf8_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'mo');
I = double(imread('rp6_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'rv');
I = double(imread('rp7_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'rs');
I = double(imread('rp8_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'ro');
I = double(imread('rp9_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'r*');
I = double(imread('np10_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'g+');
I = double(imread('np11_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'gx');
I = double(imread('np12_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'g.');
I = double(imread('np13_x10.bmp')); MNC3D1;
plot(p1(1),p2(1)/p1(1),'g<');

```

```

% I = double(imread('fc5_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'k^'); hold on;
% I = double(imread('fc6_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'kv');
% I = double(imread('fc7_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'ks');
% I = double(imread('ff5_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'b^');
% I = double(imread('ff6_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'bv');
% I = double(imread('ff7_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'bs');
% I = double(imread('rab4_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'cd');
% I = double(imread('rab5_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'c^');
% I = double(imread('rab6_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'cv');
% I = double(imread('rab7_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'cs');
% I = double(imread('rf6_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'mv');
% I = double(imread('rf7_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'ms');
% I = double(imread('rf8_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'mo');
% I = double(imread('rp6_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'rv');
% I = double(imread('rp7_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'rs');
% I = double(imread('rp8_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'ro');
% I = double(imread('rp9_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'r*');
% I = double(imread('np10_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'g+');
% I = double(imread('np11_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'gx');
% I = double(imread('np12_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'g. ');
% I = double(imread('np13_x25.bmp')); MNC3D1:
% plot(p1(1),p2(1)/p1(1),'g<');

```

Subprogram MNC3D1

```
I_a1 = zeros(256,256);
I_a2 = zeros(256,256,256);
for n = 1:1:256
    for m = 2:256
        if I(n,m) >= I(n,m-1)
            I_a1(m,I(n,m-1):I(n,m)) = 1;
        else
            I_a1(m,I(n,m):I(n,m-1)) = 1;
        end
    end
    I_a2(:,:,n) = I_a1(:,:,n);
    I_a1 = zeros(256,256);
end
MNC3D2:
% Afisare sectiuni prin suprafata 3D
% I_a3 = zeros(256,256);
% for k = 128:1:256
%     % indicii masivului I_a2
%     % primul indice = coloana: al doilea indice = inaltime:
%     % al treilea indice = linie.
%     k = 170;
%     I_a3(:,:,k) = I_a2(:,:,k);
%     imshow(imcomplement(I_a3));
% end
```

Subprogram MNC3D2

```
% indice linie
i = 1;
% indice coloana
j = 1;
% dimensiune initiala caseta
dim_cas_min = 1;
% dimensiune caseta
dim_cas = dim_cas_min;
% dimensiune finala caseta
dim_cas_max = 24;
% pasul cu care se incrementeaza dimensiunea casetei
pas_dim_cas = 1;
% numar de iteratii
nr_it = (dim_cas_max - dim_cas_min)/pas_dim_cas;
% contorizare pixeli diferiti de fundal
cont_pix_on = zeros(1,nr_it);
% dimensiunea fractala
dim_fract = zeros(1,nr_it);
% se pastreaza valorile dim_cas pe masura rularii programului
dim_cas1 = zeros(1,nr_it);
% Indicele pentru fiecare dimensiune de caseta
iteratia = 1;
% Valoarea maxima pentru fiecare dimensiune a masivului
[i_max, j_max, k_max] = size(I_a2);
% Numararea casetelor
for dim_cas = dim_cas_min: pas_dim_cas: dim_cas_max
    cont_pix_on(iteratia) = 0;
    for k = 1:dim_cas+1:k_max - dim_cas
        for i = 1:dim_cas+1:i_max - dim_cas
            for j = 1:dim_cas+1:j_max - dim_cas
                bw_cas = I_a2(i:i + dim_cas,j:j + dim_cas,k:k + dim_cas);
                if sum(sum(sum(bw_cas))) >= 1
                    cont_pix_on(iteratia) = cont_pix_on(iteratia) + 1;
                end
            end
        end
    end
    dim_cas1(iteratia) = dim_cas;
    iteratia = iteratia + 1;
end
cont_pix_on = [sum(sum(sum(I_a2))) cont_pix_on];
dim_cas1 = [0 dim_cas1] + 1;
% Calculul logaritmului
cont_pix_on2 = log10(cont_pix_on);
dim_cas2 = log10(1./dim_cas1);
% Afisare
% figure;
% plot(dim_cas2,cont_pix_on2,'k*');
```

```
% hold on
% Calcul regresie liniara
p = polyfit(dim_cas2.cont_pix_on2,1);
f = polyval(p,dim_cas2);
% Afisare aproximare
% plot(dim_cas2.f,'k');
p1 = p(1);
p2 = p(2);
```

Diagrame P_f , etaloane r_p , pentru domenii de variație a dimensiunii casetei.

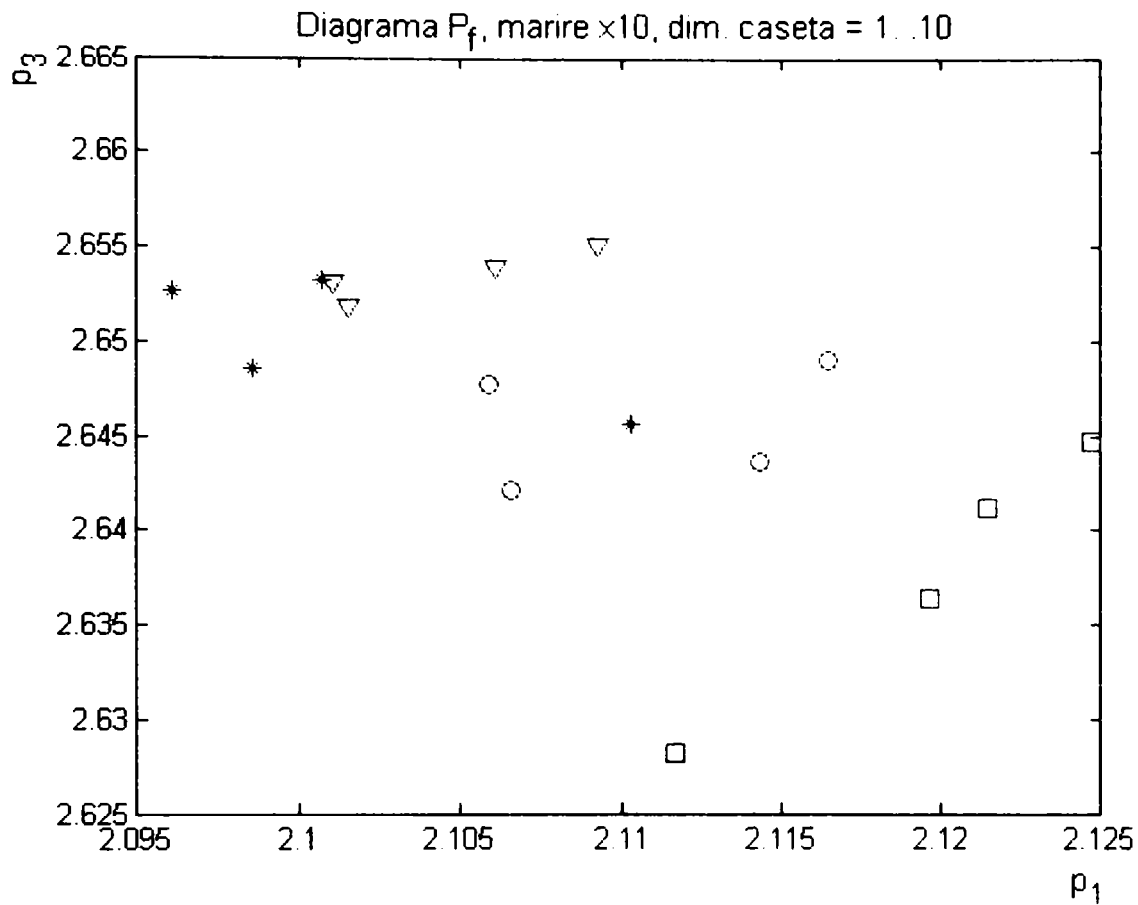


Fig.A.8.10.1. Diagrama P_f , etaloane r_p , mărime $\times 10$, dimensiunea casetei = 1..10.

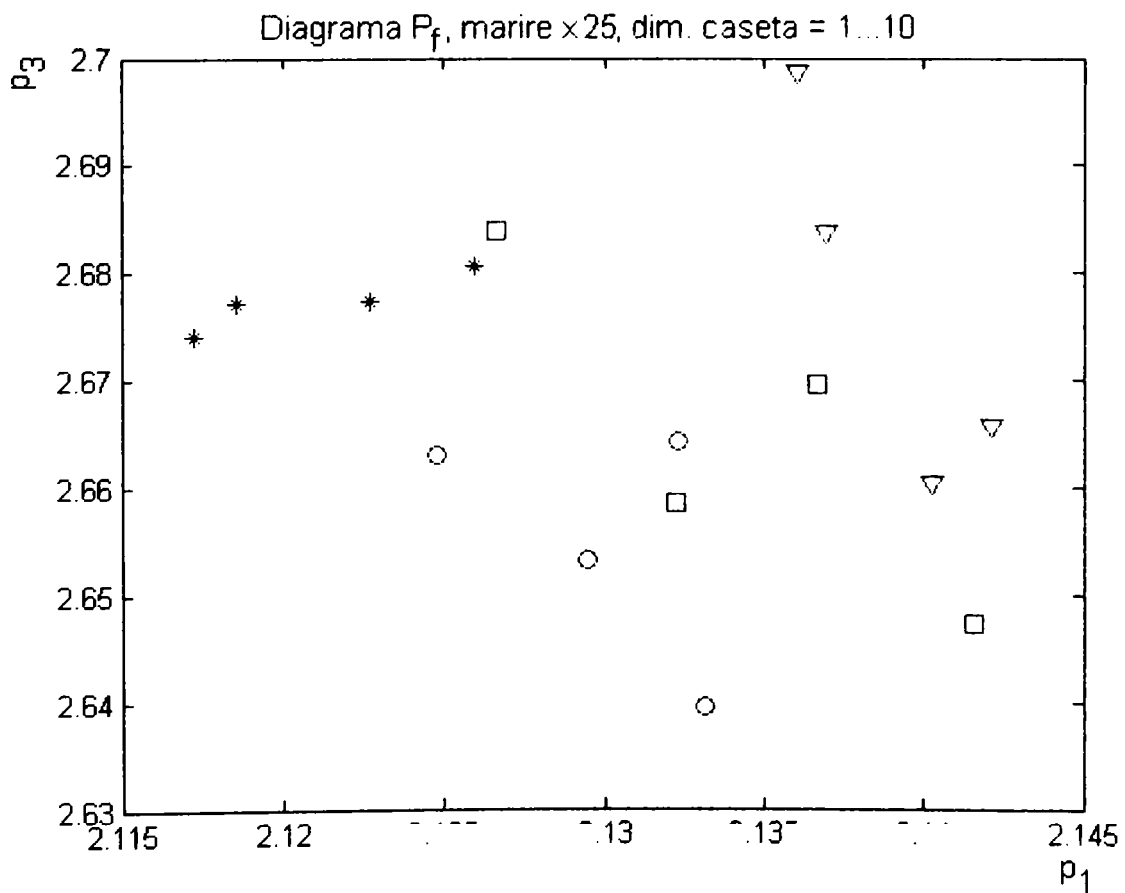


Fig.A.8.10.2. Diagrama P_f , etaloane r_p , mărime $\times 25$, dimensiunea casetei = 1..10.

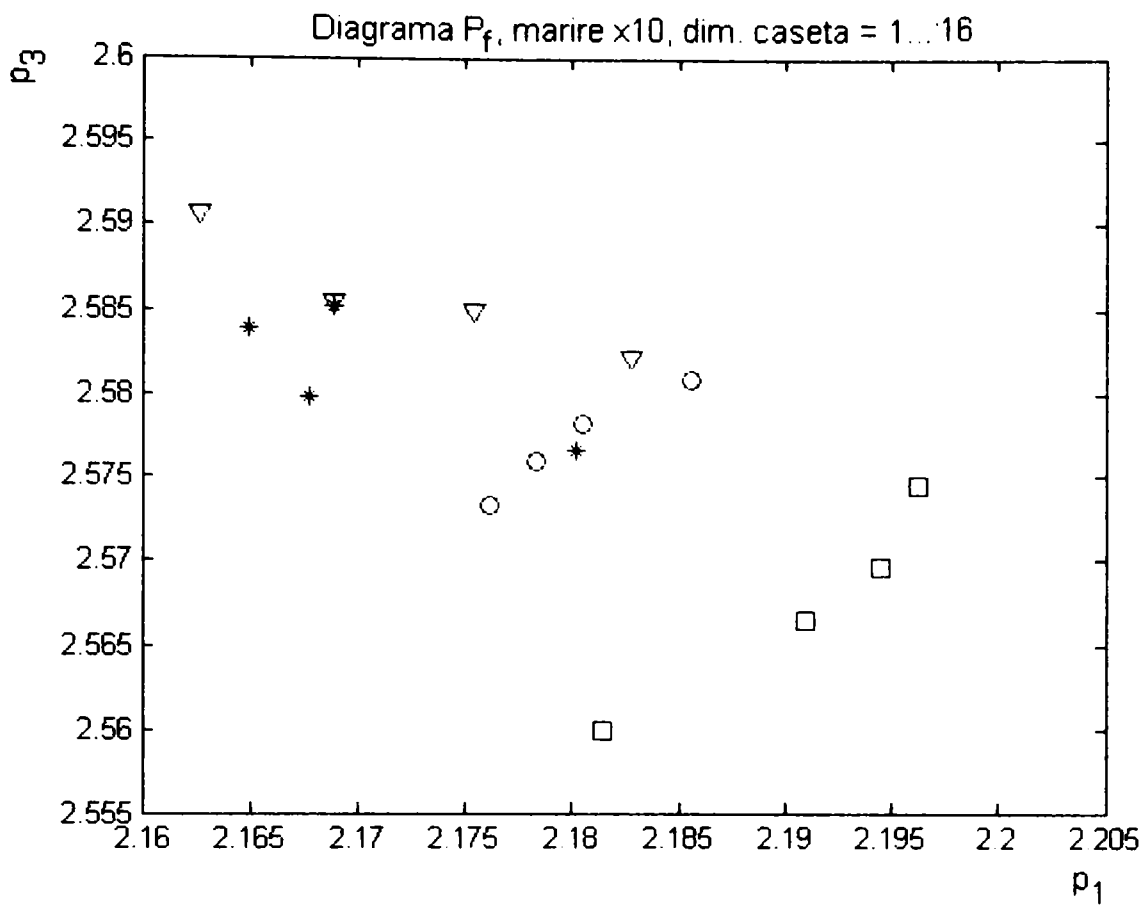


Fig.A.8.10.3. Diagrama P_f , etaloane r_p , mărire $\times 10$, dimensiunea casetei = 1..16.

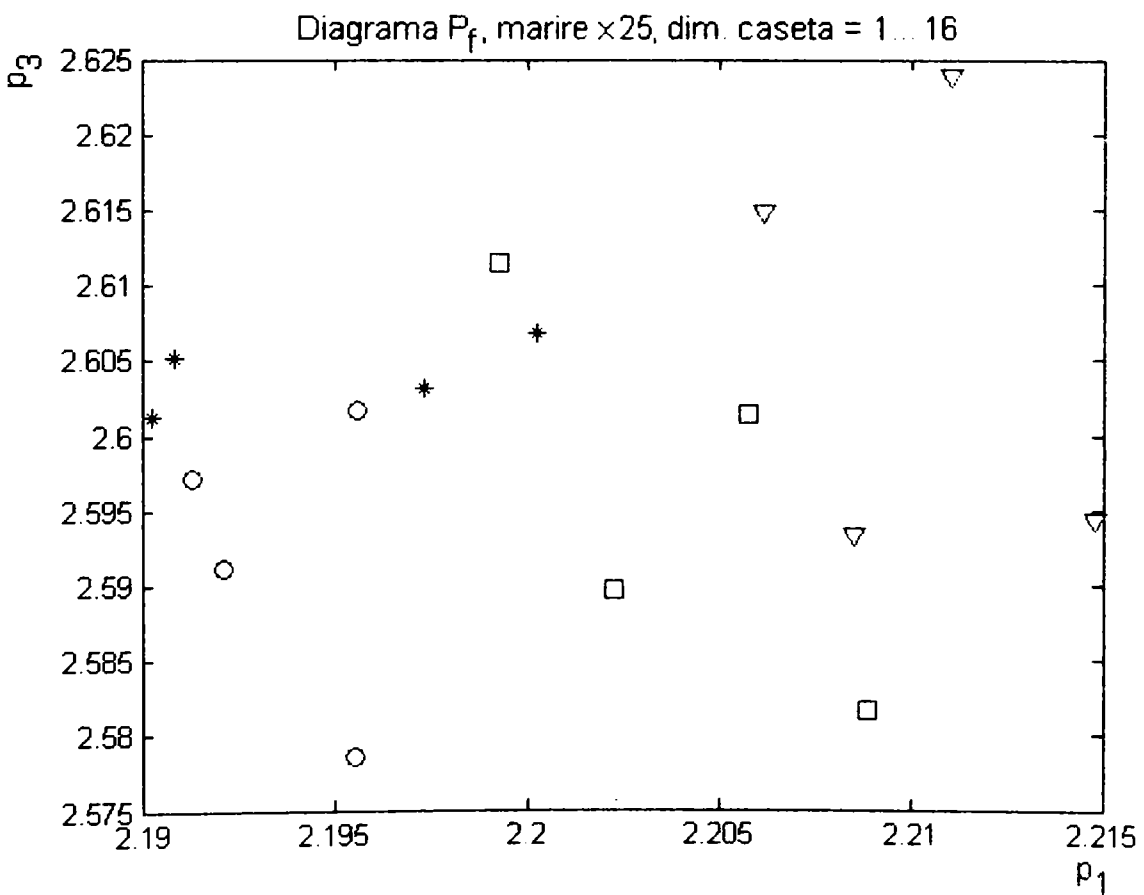


Fig.A.8.10.4. Diagrama P_f , etaloane r_p , mărire $\times 25$, dimensiunea casetei = 1..16.

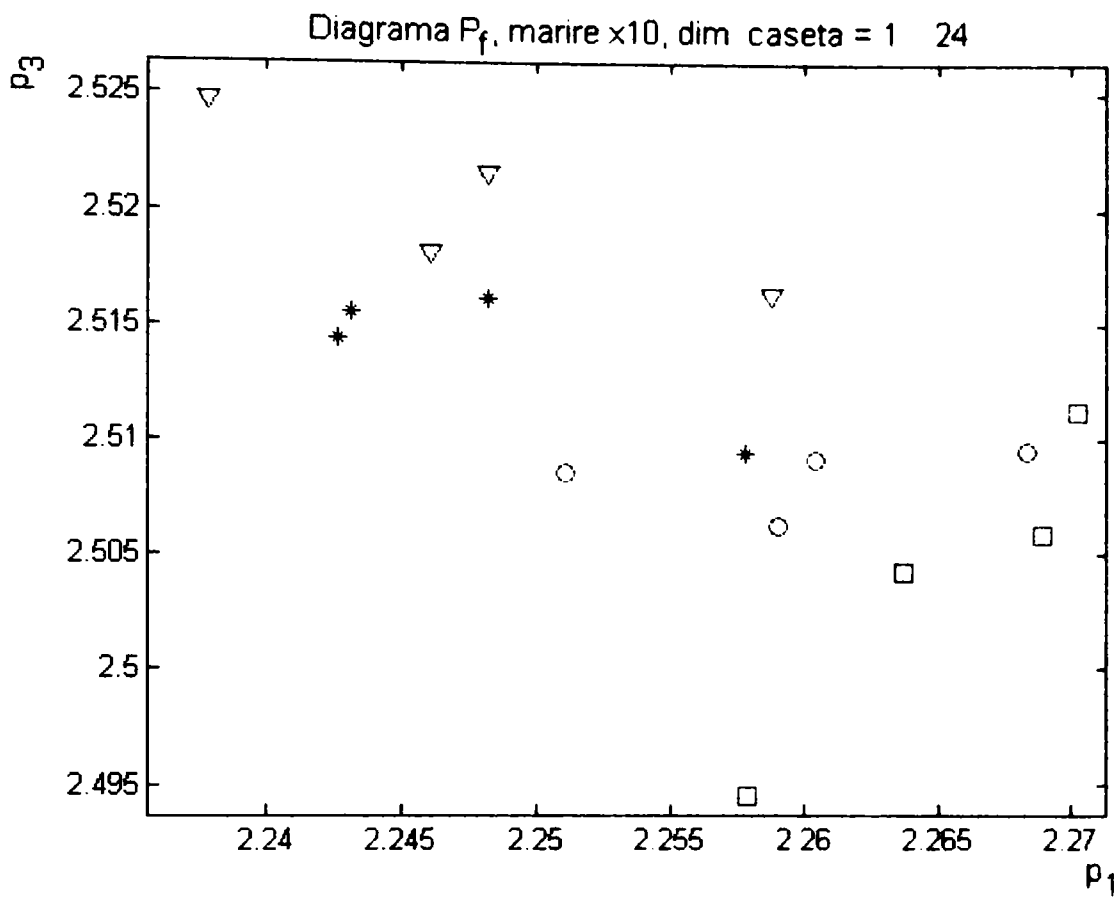


Fig.A.8.10.5. Diagrama P_f , etaloane r_p , mărire x10, dimensiunea casetei = 1..24.

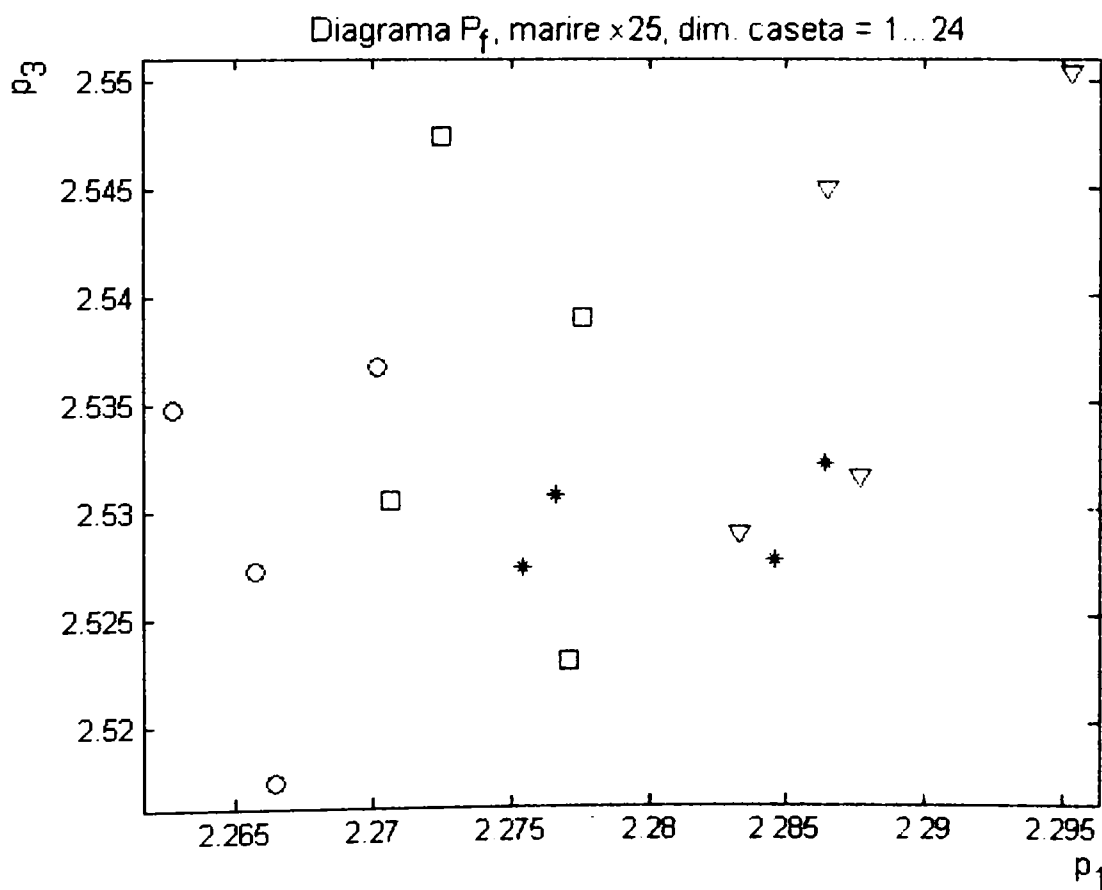


Fig.A.8.10.6. Diagrama P_f , etaloane r_p , mărire x25, dimensiunea casetei = 1..24.

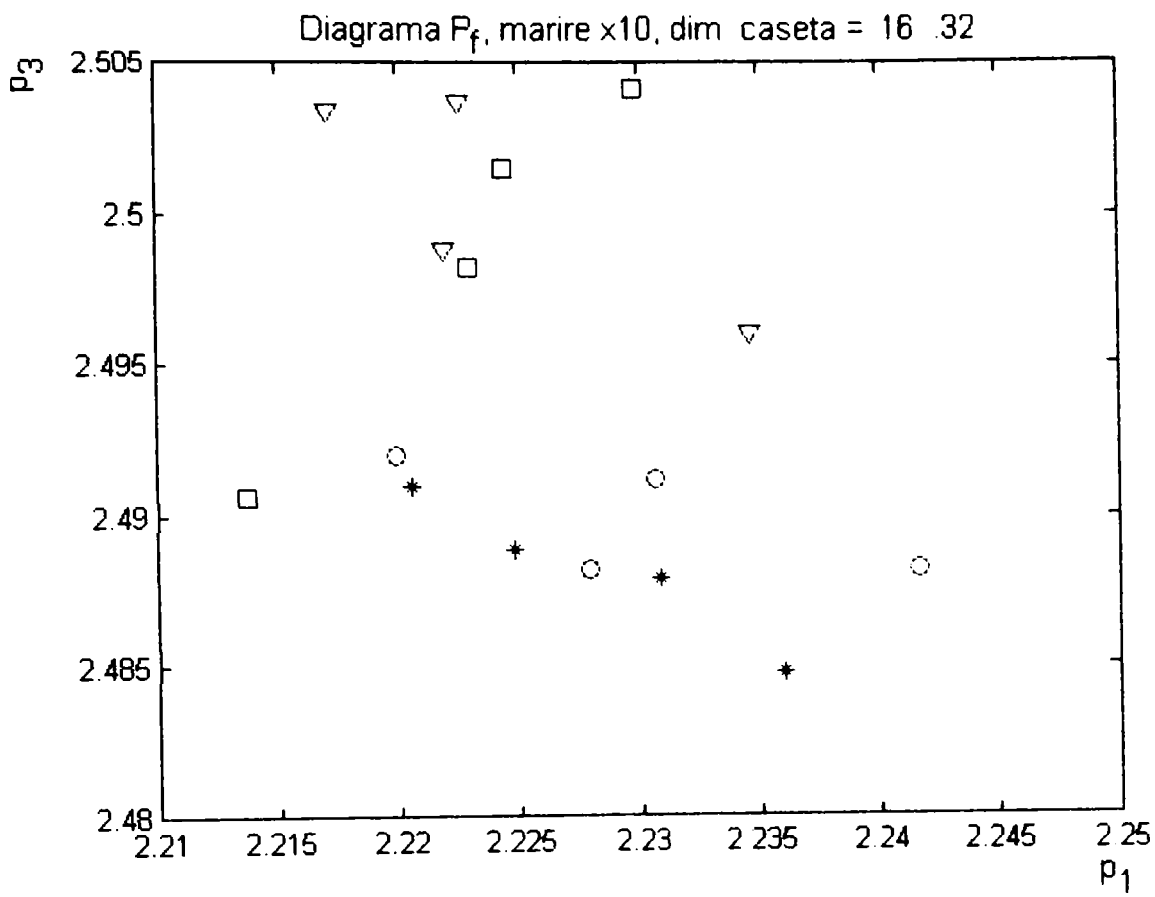


Fig.A.8.10.7. Diagrama P_f , etaloane rp, mărire x10, dimensiunea casetei = 16..32.

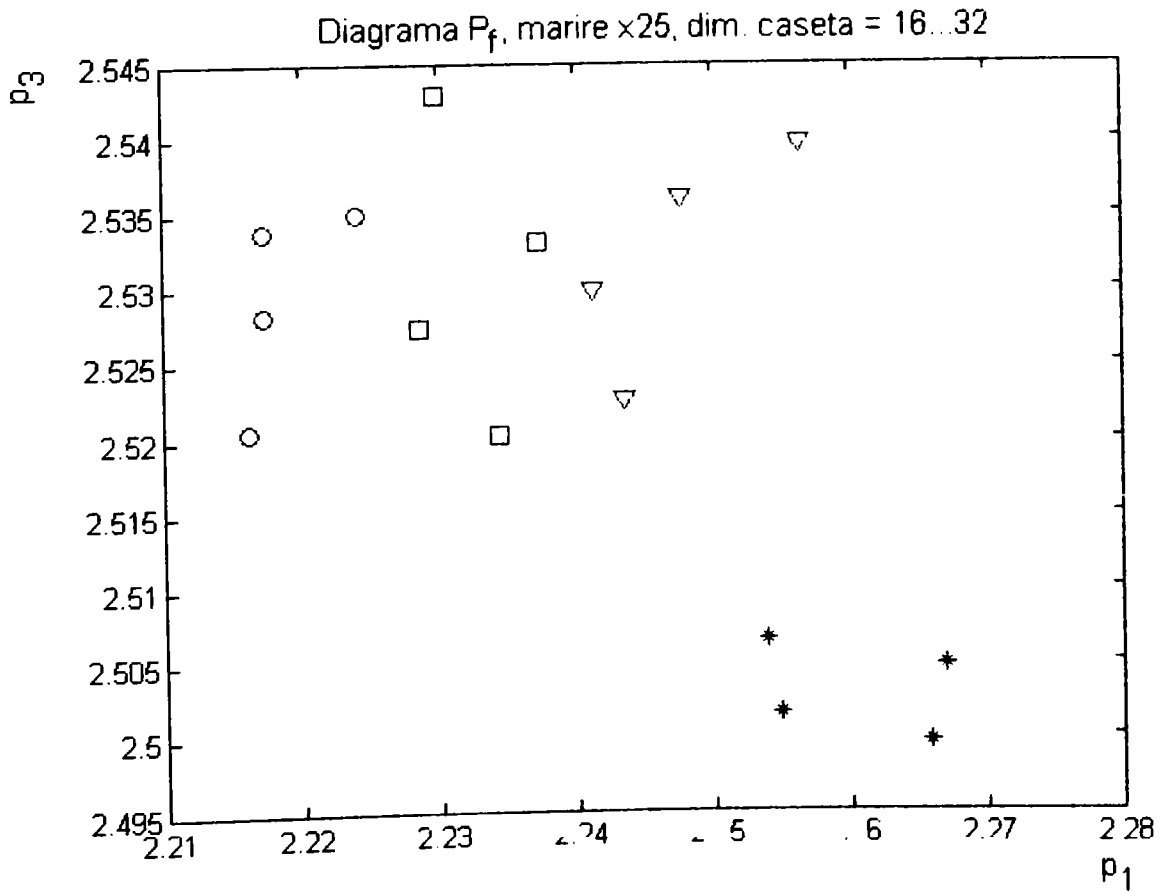


Fig.A.8.10.8. Diagrama P_f , etaloane rp, mărire x25, dimensiunea casetei = 16..32.

Program : MDSPF2Da

```

% addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\_3grad');
% addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\_3grad');
addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x10rota');
% addpath('C:\Label\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x25rota');
global p1ma p2ma p1f p2f sigmaa

k = 80;
nr_val1 = 1;
nr_val2 = k;
y0 = double(imread('fc5_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'k^'); hold on
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('fc6_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'kv');
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('fc7_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'ks');
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('ff5_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'b^'); hold on;
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('ff6_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'bv');
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('ff7_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'bs');
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('rab4_x10.bmp'));
MDSPF2Db;
plot(p1f,p2f,'cd'); hold on
rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
clear p1f p2f
y0 = double(imread('rab5_x10.bmp'));
MDSPF2Db;

```

```

    plot(p1f,p2f,'c^');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rab6_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'cv');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rab7_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'cs'); hold on;
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rf6_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'mv'); hold on;
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal;
clear p1f p2f;
    y0 = double(imread('rf7_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'ms');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rf8_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'mo');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rp6_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'rv');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rp7_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'rs');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rp8_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'ro');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('rp9_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'r*');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('np10_x10.bmp'));
    MDSPF2Db;

```

```

    plot(p1f,p2f,'g+');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('np11_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'gx');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('np12_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'g. ');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f
    y0 = double(imread('np13_x10.bmp'));
    MDSPF2Db;
    plot(p1f,p2f,'g<');
rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
clear p1f p2f

```

```

% figure;
% k = 64;
% nr_val1 = 1;
% nr_val2 = k;
% y0 = double(imread('fc5_x25.bmp'));
% MDSPF2Db; % oblio12b
% plot(p1f,p2f,'k^'); hold on
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('fc6_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'kv');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('fc7_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'ks');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('ff5_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'b^'); hold on
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('ff6_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'bv');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('ff7_x25.bmp'));
% MDSPF2Db;

```

```

% plot(p1f,p2f,'bs');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rab4_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'cd'); hold on
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rab5_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'c^');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rab6_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'cv');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rab7_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'cs'); hold on;
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rf6_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'mv'); hold on
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rf7_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'ms');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rf8_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'mo');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rp6_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'rv');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rp7_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'rs');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa], 'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rp8_x25.bmp'));
% MDSPF2Db;

```

```

% plot(p1f,p2f,'ro');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('rp9_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'r*');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('np10_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'g+');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('np11_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'gx');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('np12_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'g. ');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f
% y0 = double(imread('np13_x25.bmp'));
% MDSPF2Db;
% plot(p1f,p2f,'g<');
% rectangle('Position', [p1ma p2ma sigmaa sigmaa],'Curvature', [1 1]); axis equal
% clear p1f p2f

```

Subprogram MDSPF2Db

```

f = (nr_val1:nr_val2);
% Prelucrare pe linii
y01 = y0 - repmat(mean(y0.2),1,256);
y02 = y01;
% Transformata Fourier pe orizontala
Y1 = fft(y02,256,2);
% Parametrii fractali
Pyy1 = Y1.*conj(Y1) / 256 + 0.00001;
Pyy1 = Pyy1(:,nr_val1:nr_val2);
flog = log10(f);
Pyy1log = log10(Pyy1);
for n = 1:256
    p10 = polyfit(flog,Pyy1log(n,:),1);
    f_Pyy1(n,nr_val1:nr_val2) = polyval(p10,flog);
    p1(n,:) = p10;
end
p1m = mean(p1(:,1));
p1s = std(p1(:,1));

```

```

f = (nr_val1:nr_val2);
% Transformata Fourier pe verticala
Y2 = fft(y02,256,1);
% Parametrii fractali
Pyy2 = Y2.*conj(Y2) / 256 + 0.00001;
Pyy2 = Pyy2(nr_val1:nr_val2,:);
flog = log10(f);
Pyy2log = log10(Pyy2);
for n = 1:256
    p20 = polyfit(flog,Pyy2log(n,:),1);
    f_Pyy2(n,nr_val1:nr_val2) = polyval(p20,flog);
    p2(n,:) = p20;
end
p2m = mean(p2(:,1));
p2s = std(p2(:,1));

sigma = min([p1s p2s]);
j = 1;
for nk = 1:256
    p1c = abs(p1(nk,1) - p1m);
    p2c = abs(p2(nk,1) - p2m);
    raza_p = sqrt(abs(p1c^2 + p2c^2));
    if raza_p <= sigma
        p1f(j) = p1(nk,1);
        p2f(j) = p2(nk,1);
        j = j + 1;
    end
end
p1ma = p1m - sigma;
p2ma = p2m - sigma;
sigmaa = 2*sigma;

```


Program : DIFa

```
% addpath('C:\Labcel\Teza final\Cap 7 Analiza suprafata\Programe final\_3grad');
% addpath('C:\Labcel\Teza final\Cap 7 Analiza suprafata\Programe final\3grad');
  addpath('C:\Labcel\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x10rota');
% addpath('C:\Labcel\Teza final\Cap 7 Analiza suprafata\Programe final\Imagini x25rota');
global p1ma p2ma p1f p2f sigmaa
```

```
figure:
```

```
k = 120;
```

```
nr_val1 = 1;
```

```
nr_val2 = k;
```

```
  y0 = double(imread('fc5_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'k^'); hold on
```

```
  y0 = double(imread('fc6_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'kv');
```

```
  y0 = double(imread('fc7_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'ks');
```

```
  y0 = double(imread('ff5_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'b^'); hold on;
```

```
  y0 = double(imread('ff6_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'bv');
```

```
  y0 = double(imread('ff7_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'bs');
```

```
  y0 = double(imread('rab4_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'cd'); hold on
```

```
  y0 = double(imread('rab5_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'c^');
```

```
  y0 = double(imread('rab6_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'cv');
```

```
  y0 = double(imread('rab7_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'cs'); hold on;
```

```
  y0 = double(imread('rf6_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'mv'); hold on
```

```
  y0 = double(imread('rf7_x10.bmp'));
```

```
  DIFb;
```

```
  plot(p1,p2,'ms');
```

```

y0 = double(imread('rf8_x10.bmp'));
DIFb;
plot(p1,p2.'mo');
y0 = double(imread('rp6_x10.bmp'));
DIFb;
plot(p1,p2.'rv');
y0 = double(imread('rp7_x10.bmp'));
DIFb;
plot(p1,p2.'rs');
y0 = double(imread('rp8_x10.bmp'));
DIFb;
plot(p1,p2.'ro');
y0 = double(imread('rp9_x10.bmp'));
DIFb;
plot(p1,p2.'r*');
y0 = double(imread('np10_x10.bmp'));
DIFb;
plot(p1,p2.'g+');
y0 = double(imread('np11_x10.bmp'));
DIFb;
plot(p1,p2.'gx');
y0 = double(imread('np12_x10.bmp'));
DIFb;
plot(p1,p2.'g. ');
y0 = double(imread('np13_x10.bmp'));
DIFb;
plot(p1,p2.'g<');

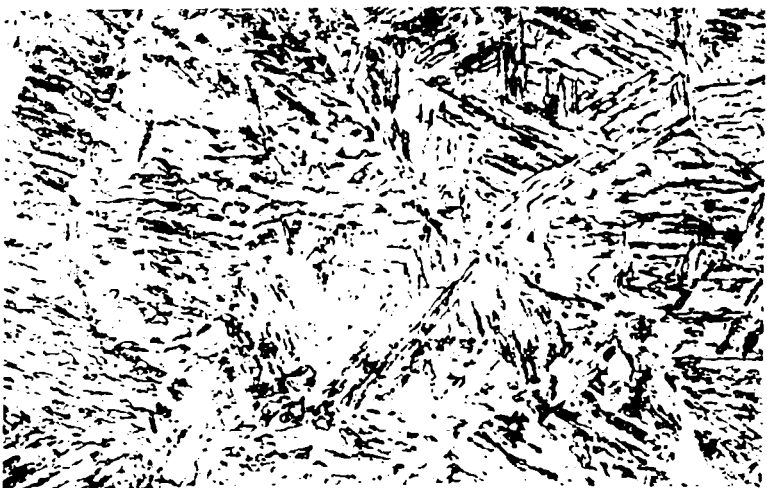
```

Subprogram DIFb

```

f = (nr_val1:nr_val2);
% Prelucrare pe linii
y01 = y0 - repmat(mean(y0,2),1,256);
y02 = y01;
% Transformata Fourier pe orizontala
Y1 = fft(y02,256,2);
% Parametrii fractali
Pyy1 = Y1.*conj(Y1) / 256 + 0.00001;
Pyy1 = Pyy1(:,nr_val1:nr_val2);
Pyy1sum1 = mean(Pyy1);
y02sum = mean(y02);
Y1sum = fft(y02sum,256);
Pyy1sum2 = Y1sum.*conj(Y1sum) / 256 + 0.00001;
Pyy1sum2 = Pyy1sum2(nr_val1:nr_val2);
Pyy1dif = (Pyy1sum1 - Pyy1sum2);
flog = log10(f);
Pyy1log1 = log10(Pyy1dif);
p1 = polyfit(flog,Pyy1log1,1);
p1 = p1(1);
f_Pyy1(nr_val1:nr_val2) = polyval(p1,flog);
p2 = log10(max(Pyy1dif));

```

Baza de imagini utilizată pentru analizele de tip A.*Microstructuri cu aspect acicular.***Fig.A.9.1.1.** Bainită inferioară, mărire x500.**Fig.A.9.1.2.** Bainită superioară, mărire x500.**Fig.A.9.1.3.** Martensită de călire, mărire x500.

Microstructuri cu grafit lamelar



Fig.A.9.1.4. Fontă cenușie feritică. mărire x500.



Fig.A.9.1.5. Fontă cenușie ferito-perlitică. mărire x500.



Fig.A.9.1.6. Fontă cenușie perlitică. mărire x500.

Microstructuri cu grafit nodular

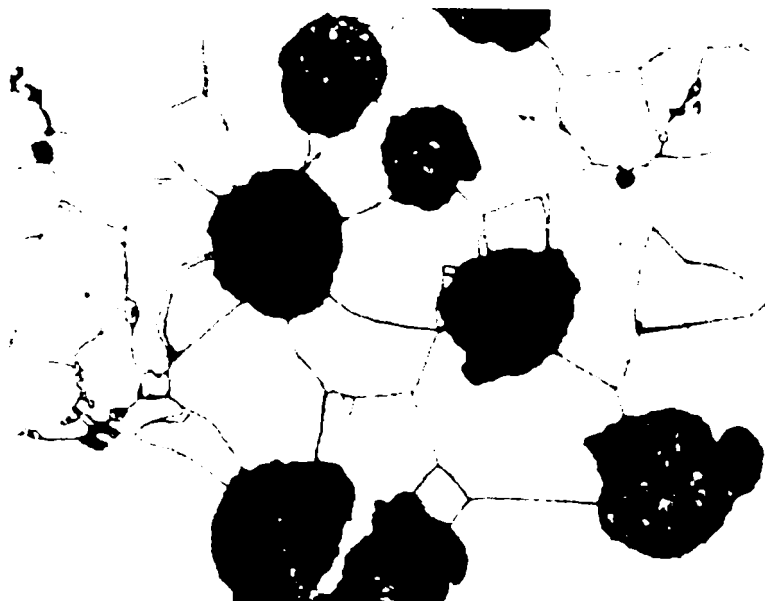


Fig.A.9.1.7. Fontă nodulară feritică mărire x300.



Fig.A.9.1.8. Fontă nodulară ferito-perlitică. mărire x300.



Fig.A.9.1.9. Fontă nodulară perlitică. mărire x300.

Microstructuri ledeburitice

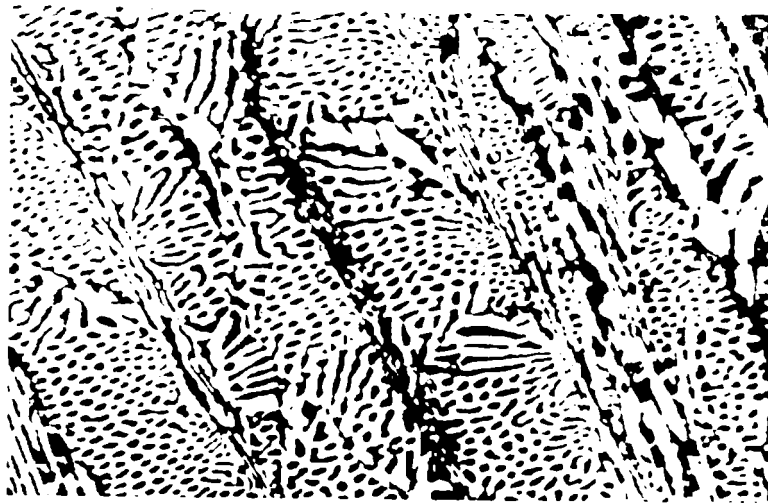


Fig.A.9.1.10. Fontă albă 4.3%C, ledeburită, mărire x500.

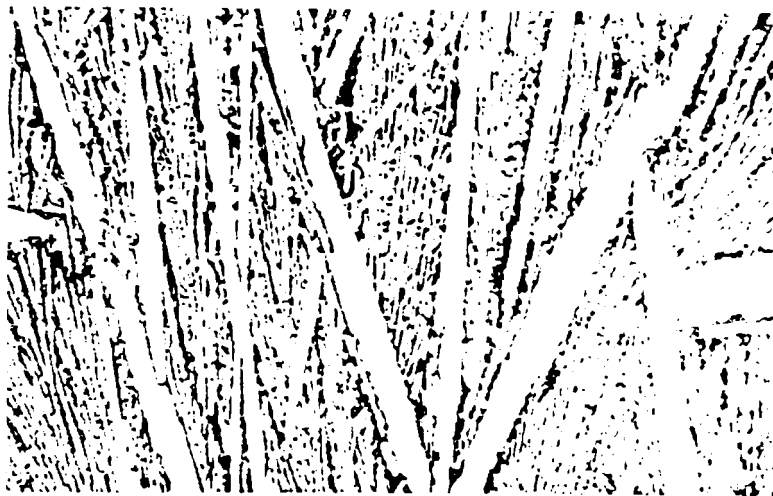


Fig.A.9.1.11. Fontă albă 5%C, ledeburită + cementită primară, mărire x500.

Microstructuri perlitice



Fig.A.9.1.12. Bainită inferioară revenită, mărire x500.



Fig.A.9.1.13. Bainită superioară revenită. mărire x500.

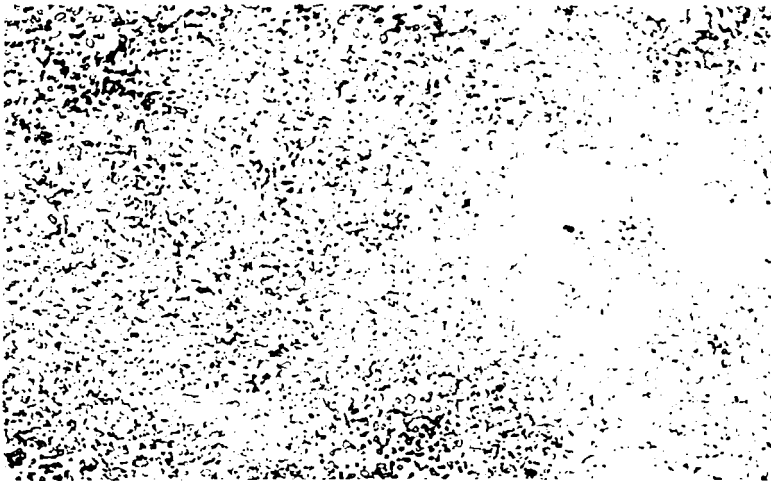


Fig.A.9.1.14. Perlită globulară. mărire x500.

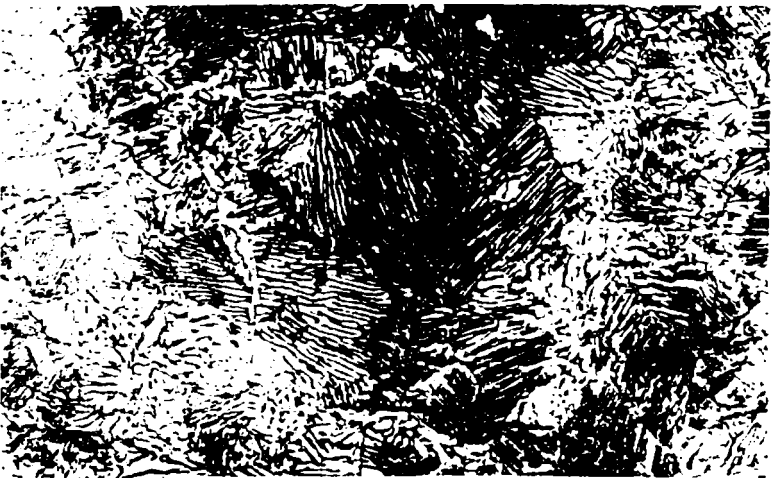


Fig.A.9.1.15. Perlită lamelară. mărire x500.

Microstructuri tip grăunte

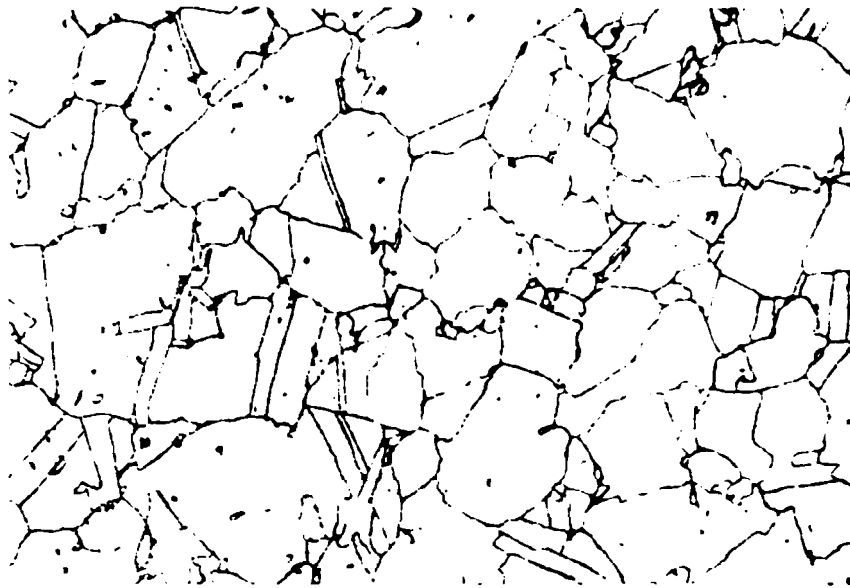


Fig.A.9.1.16. Austenită, mărire x300.

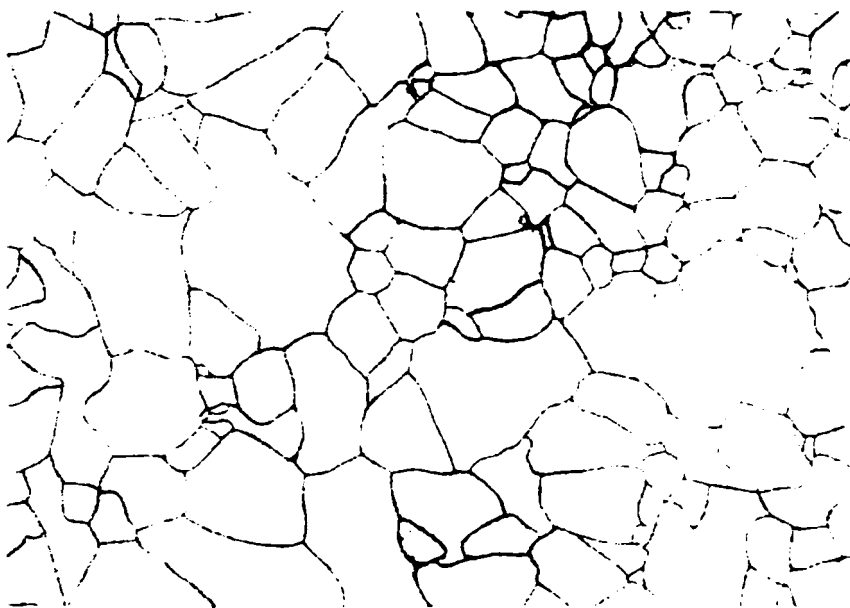


Fig.A.9.1.17. Ferită, mărire x300.

Tabel A.9.1.1

Tip microstructură	Mărire	Denumire	Simbol utilizat în diagrame
Microstructuri cu aspect acicular	x500	Bainită inferioară	○
		Bainită superioară	×
		Martensită de călire	+
Microstructuri cu grafit lamelar	x500	Fontă cenușie feritică	○
		Fontă cenușie ferito-perlitică	×
		Fontă cenușie perlitică	+
Microstructuri cu grafit nodular	x300	Fontă nodulară feritică	○
		Fontă nodulară ferito-perlitică	×
		Fontă nodulară perlitică	+
Microstructuri ledeburitice	x500	Fontă albă 4.3%C, ledeburită	○
		Fontă albă 5%C, ledeburită + cementită primară	×
Microstructuri perlitice	x500	Bainită de revenire inferioară	○
		Bainită de revenire superioară	×
		Perlită globulară	+
		Perlită lamelară	*
Microstructuri tip grăunte	x300	Austenită	○
		Ferită	×

Diagrame rezultate în urma testării programului „Identificare_A”.

Structuri aciculare

$$k_n = 1 \dots 255, i_{hmed} = 20$$

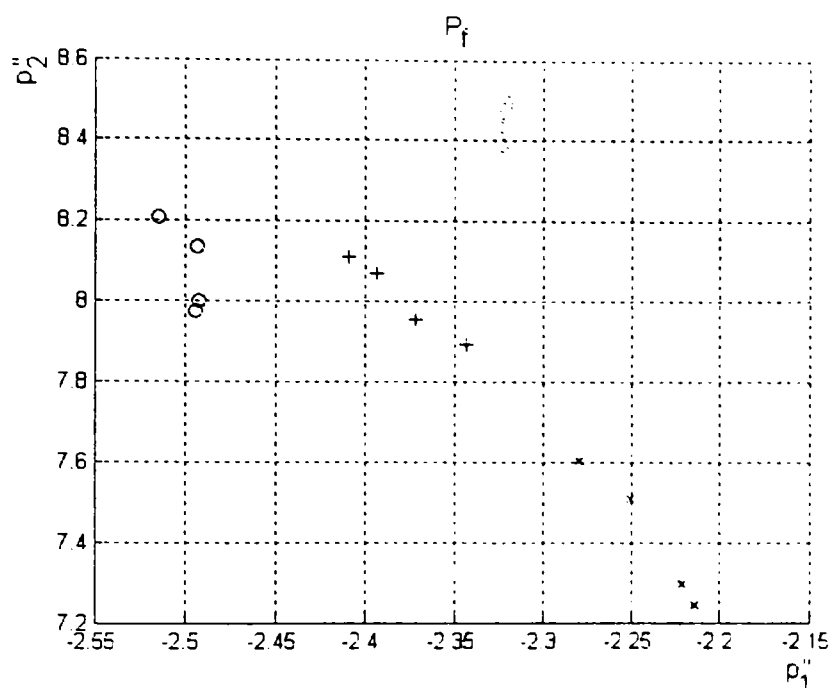


Fig.A.9.1.18 Diagrama P_f pentru structuri aciculare.

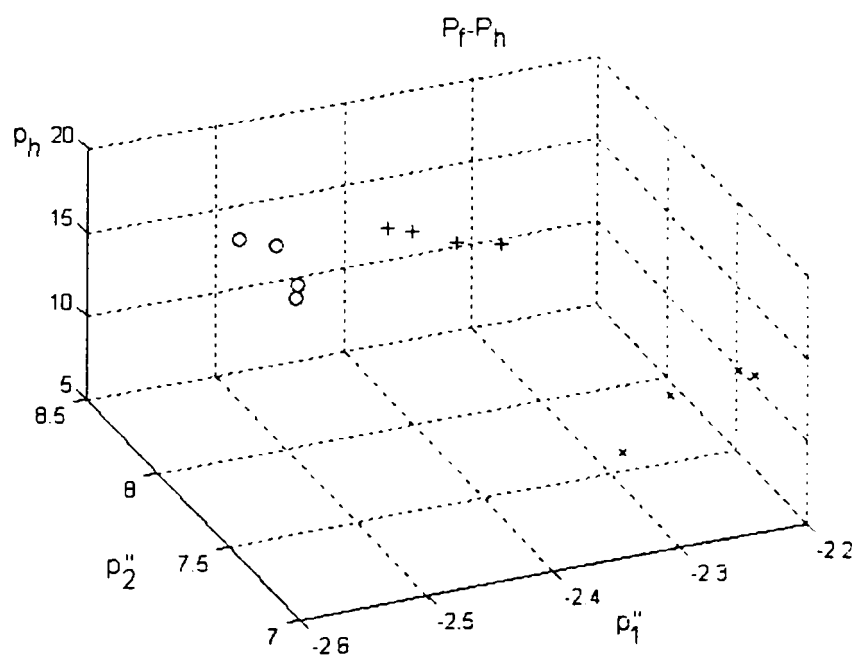


Fig.A.9.1.19 Diagrama P_f-P_h pentru structuri aciculare.

Structuri cu grafit lamelar

$$k_n = 1 \dots 255, i_{hmed} = 20$$

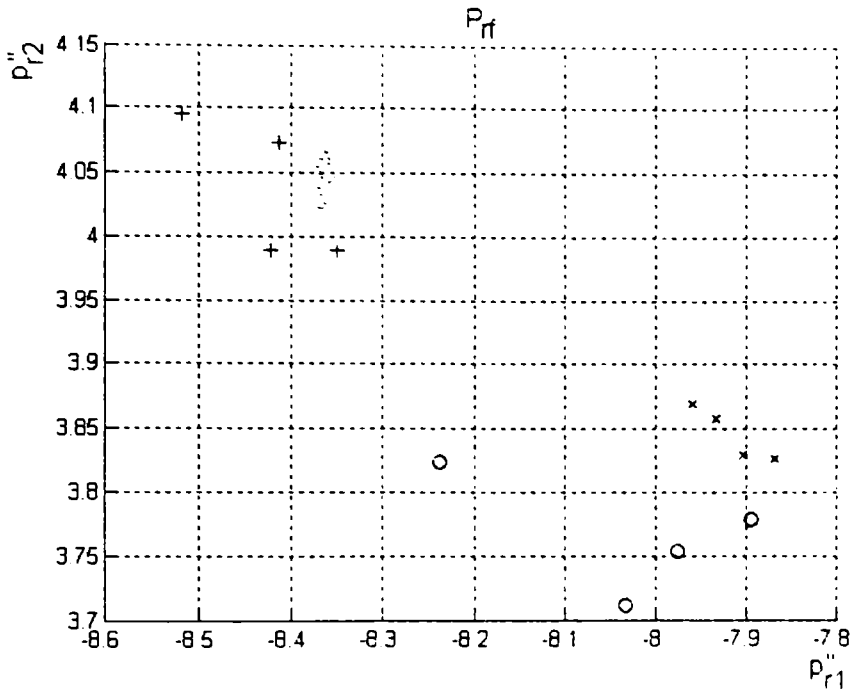


Fig.A.9.1.20 Diagrama P_{rf} pentru structuri cu grafit lamelar.

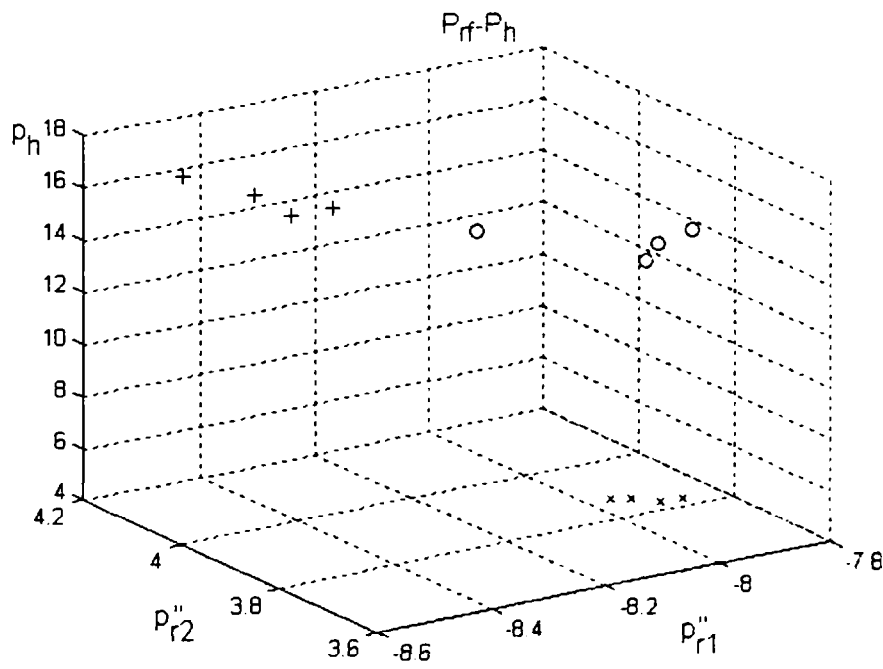


Fig.A.9.1.21 Diagrama $P_{rf}-P_h$ pentru structuri cu grafit lamelar.

Structuri cu grafit nodular

$$k_n = 16 \dots 96, i_{hmed} = 20$$

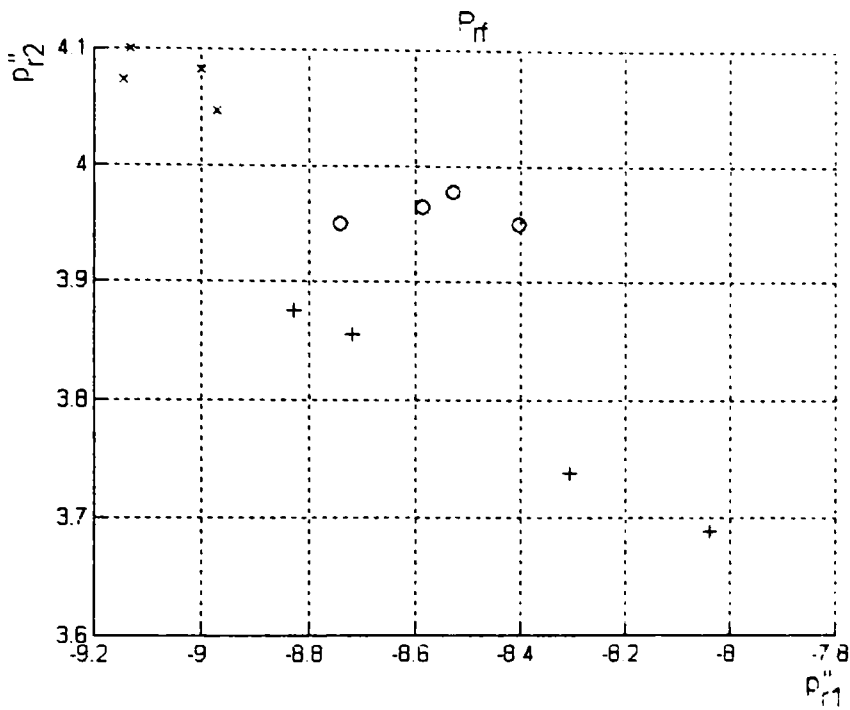


Fig.A.9.1.22 Diagrama P_{rf} pentru structuri cu grafit nodular.

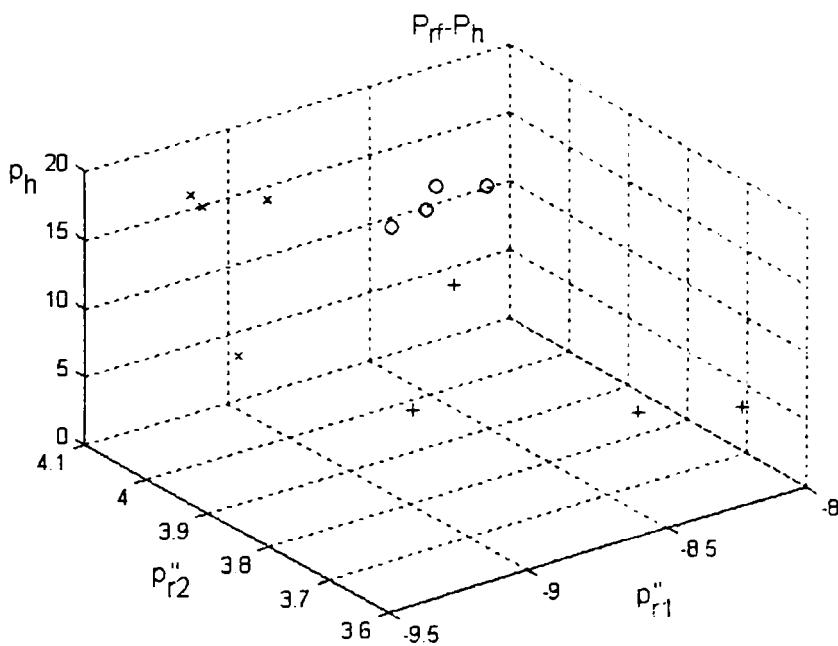


Fig.A.9.1.23 Diagrama $P_{rf}-P_h$ pentru structuri cu grafit nodular.

Structuri ledeburitice

$$k_n = 1 \dots 255, i_{hmed} = 20$$

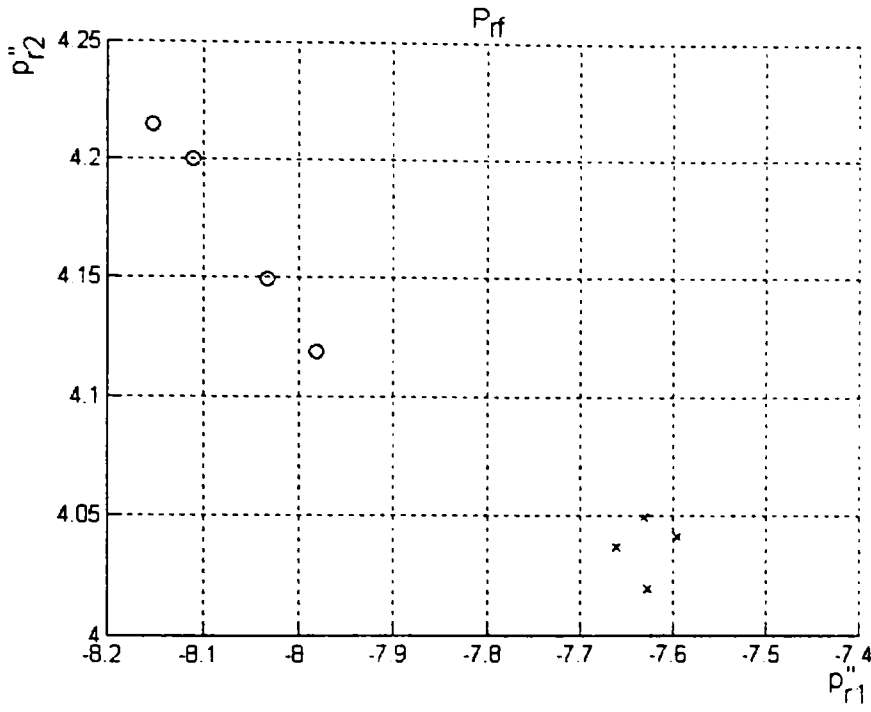


Fig.A.9.1.24 Diagrama P_{rf} pentru structuri ledeburitice.

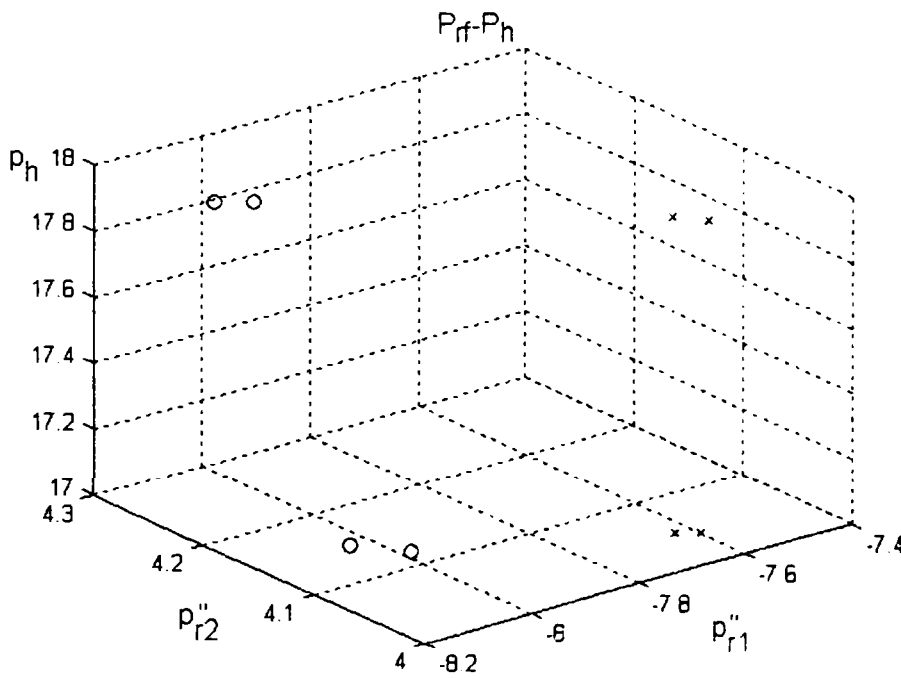


Fig.A.9.1.25 Diagrama $P_{r1}-P_h$ pentru structuri ledeburitice.

Structuri perlitice

$$k_n = 1 \dots 255, i_{hmed} = 20$$

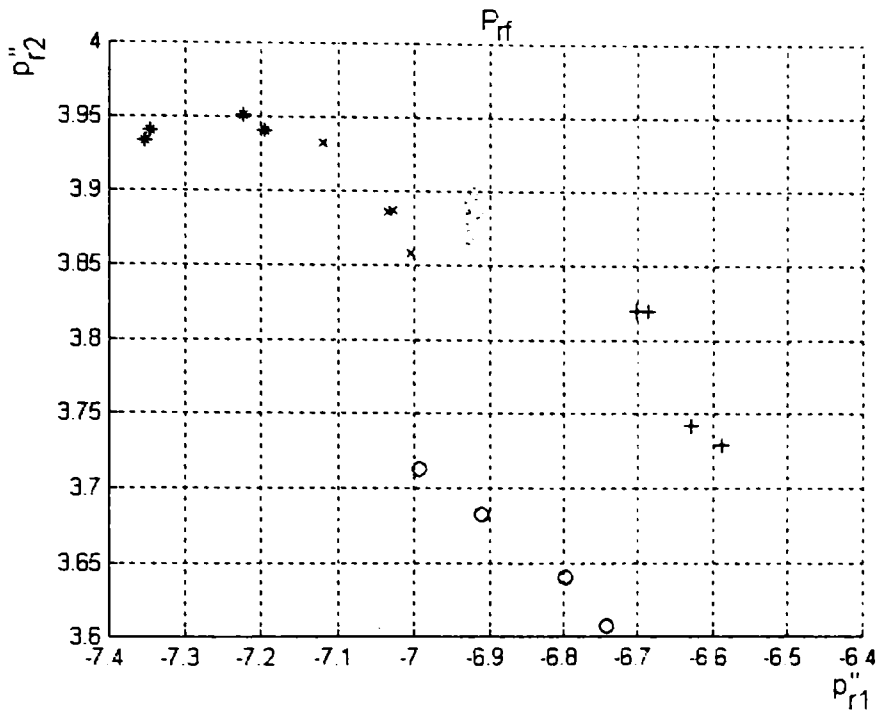


Fig.A.9.1.26 Diagrama P''_{r1} pentru structuri perlitice.

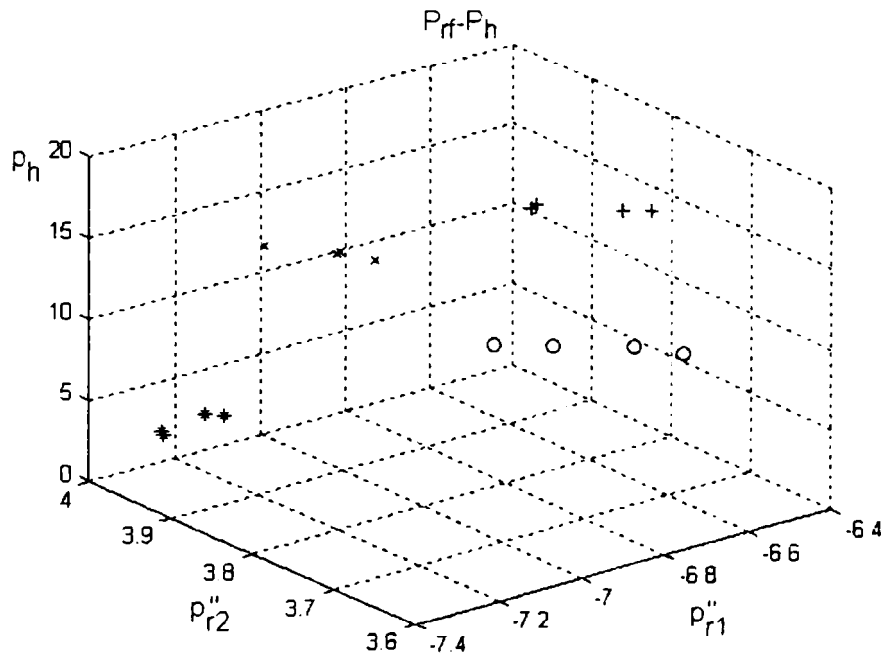


Fig.A.9.1.27 Diagrama P''_{r1} - P_h pentru structuri perlitice.

Structuri tip grăunte

$$k_n = 1 \dots 255, i_{hmed} = 20$$

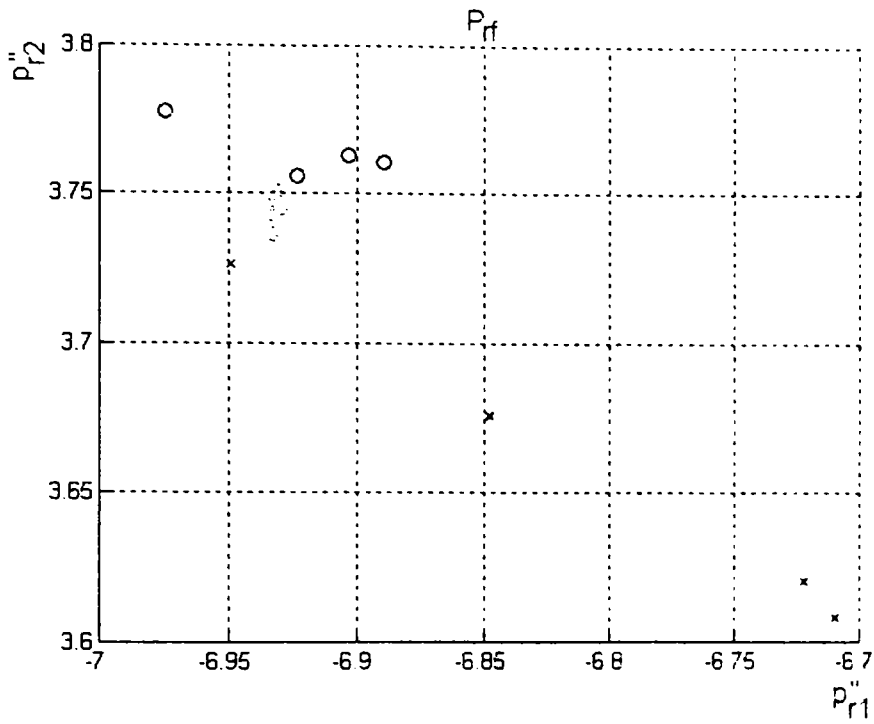


Fig.A.9.1.28 Diagrama P_{rf} pentru structuri tip grăunte.

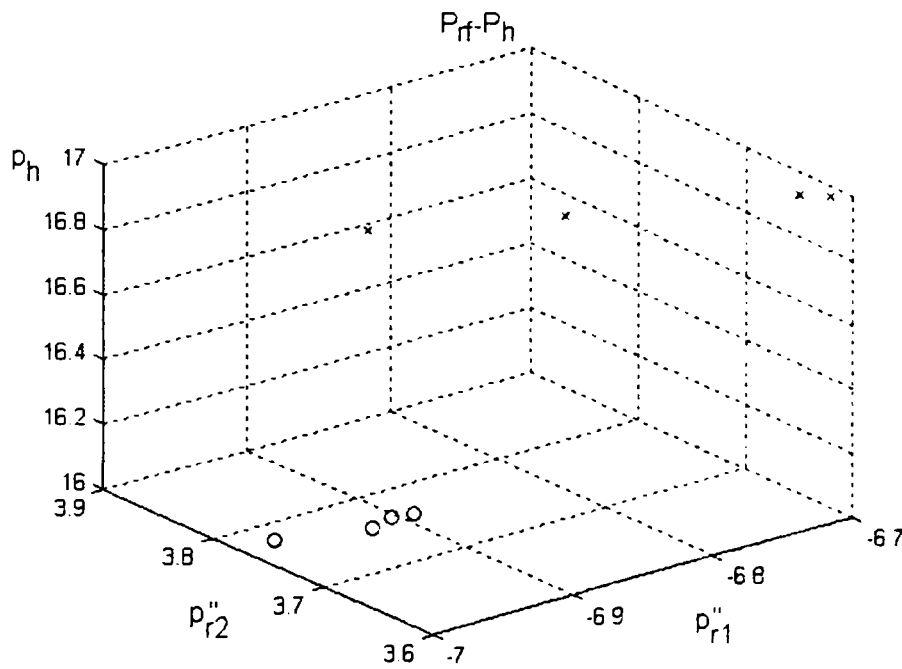


Fig.A.9.1.29 Diagrama $P_{rf}-P_h$ pentru structuri tip grăunte.

**Program de identificare a microstructurilor în cadrul analizelor de tip
A : Identificare_A.**

```

global p1 p2
figure:
k = 255;
nr_val1 = 1;
nr_val2 = k;
% 1. Structuri aciculare
addpath('C:\Label\Teza\final\final\Cap9Recunoasterestructurii\Bazadeimagini\Structuriaciculare
\');
y0 = double(imread('bainita_inferioara_x500_1.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'k^'); hold on
y0 = double(imread('bainita_inferioara_x500_2.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'kv'); hold on
y0 = double(imread('bainita_inferioara_x500_3.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'ks'); hold on
y0 = double(imread('bainita_inferioara_x500_4.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'ko'); hold on
y0 = double(imread('bainita_superioara_x500_1.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'b^'); hold on
y0 = double(imread('bainita_superioara_x500_2.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'bv'); hold on
y0 = double(imread('bainita_superioara_x500_3.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'bs'); hold on
y0 = double(imread('bainita_superioara_x500_4.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'bo'); hold on
y0 = double(imread('martensita_de_calire_x500_1.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'r^'); hold on
y0 = double(imread('martensita_de_calire_x500_2.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'rv'); hold on
y0 = double(imread('martensita_de_calire_x500_3.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'rs'); hold on
y0 = double(imread('martensita_de_calire_x500_4.bmp'));
ident_fft2Dsuma;
plot3(p1,p2,imh.'ro'); hold on
grid on

```


Subprogram de calcul a parametrilor fractali : ident_fft2Dsuma

```
dimensx = (1:512);
x = reshape(repmat(dimensx,512,1),512*512,1);
dimensy = (1:512)';
y = repmat(dimensy,512,1);
x = x - 256;
y = y - 256;
Y = fft2(y0);
Z = fftshift(Y);
Pyy2 = (((Z.*conj(Z))/(512*512))));
z = reshape(Pyy2,512*512,1);
r = round(sqrt(x.^2 + y.^2));
pcart = [r z];
pcart_r = sortrows(pcart,1);
lim = (find(pcart_r(:,1)<256));
lim2 = lim(end);
pcart_r = pcart_r(1:lim2,:);
for i = 1:255
    r_1 = find(pcart_r(:,1) == i);
    lim1_1 = r_1(1);
    lim1_2 = r_1(end);
    val_r_suma(i) = sum(pcart_r(lim1_1:lim1_2,2))/(lim1_2-lim1_1);
    x1(i) = i;
end
x_log = log10(x1);
val_r_suma_log = log10(val_r_suma);
p0 = polyfit((x_log(nr_val1:nr_val2)),(val_r_suma_log(nr_val1:nr_val2)),1);
f_val = polyval(p0,x1);
p1 = p0(1);
p2 = p0(2);
% Histograma
bins = 20;
[counts.x] = imhist(uint8(y0),bins);
[m imh] = max(counts);
```

Baza de imagini utilizată pentru analize de tip B.

Microstructuri pentru oțel 0,08-0,15%C.



Fig.A.9.3.1.Oțel 0.08-0.15%C, laminat la cald, mărire x500.

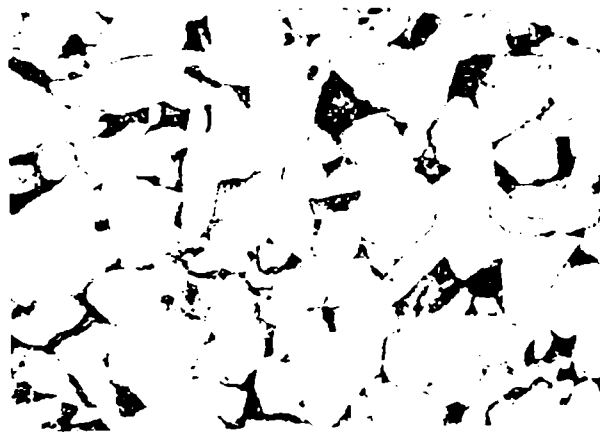


Fig.A.9.3.2.Oțel 0.08-0.15%C, normalizat, mărire x500.

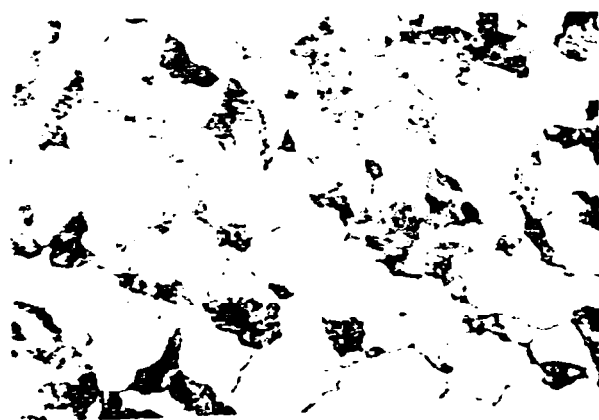


Fig.A.9.3.3.Oțel 0.08-0.15%C, recopt, mărire x500.

Microstructuri pentru oțel 0,16-0,25% C.



Fig.A.9.3.4.Oțel 0,16-0,25% C, laminat la cald, mărire x500.

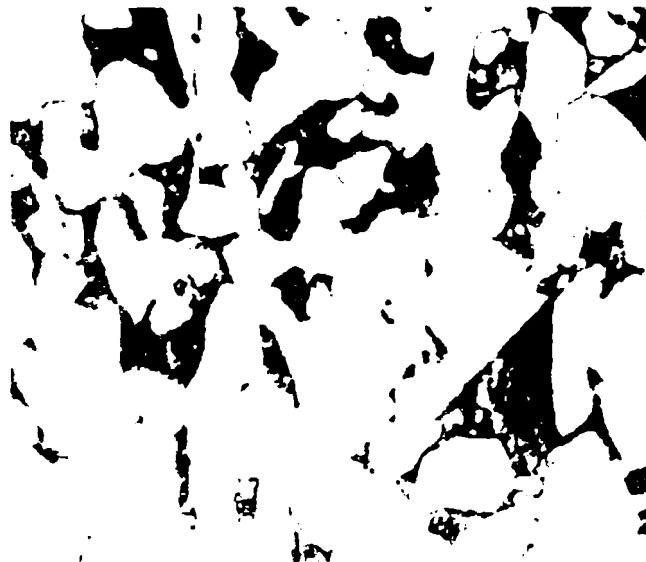


Fig.A.9.3.5.Oțel 0,16-0,25% C, normalizat, mărire x500.

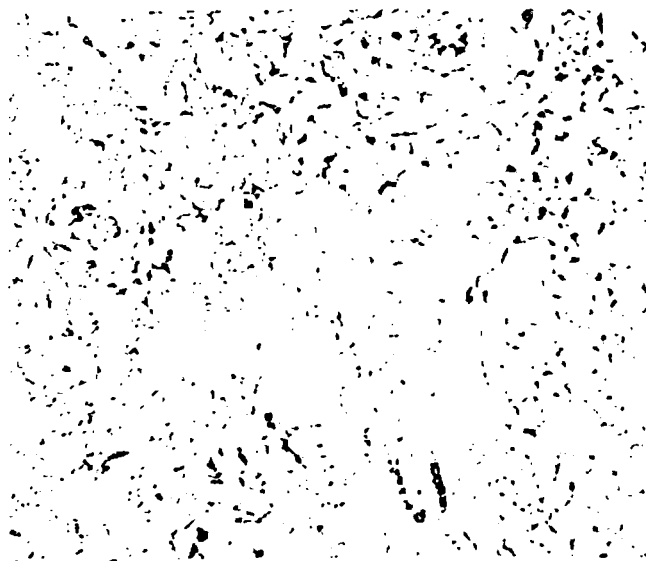


Fig.A.9.3.6.Oțel 0,16-0,25% C, recopt, mărire x500.

Microstructuri pentru oțel 0,3-0,4%C.



Fig.A.9.3.7.Oțel 0.3-0.4% C, laminat la cald, mărire x500.

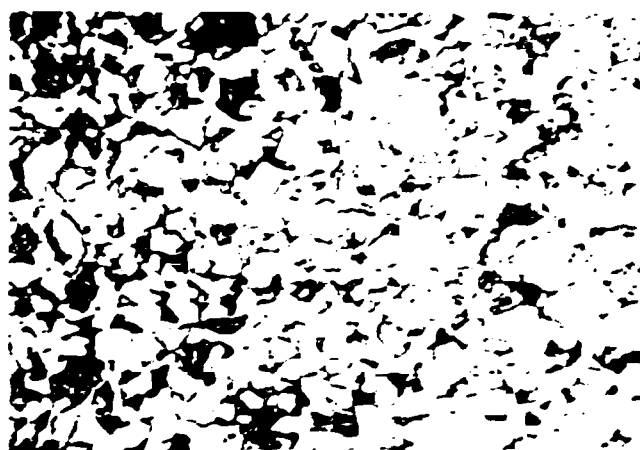


Fig.A.9.3.8.Oțel 0.3-0.4% C, normalizat, mărire x500.

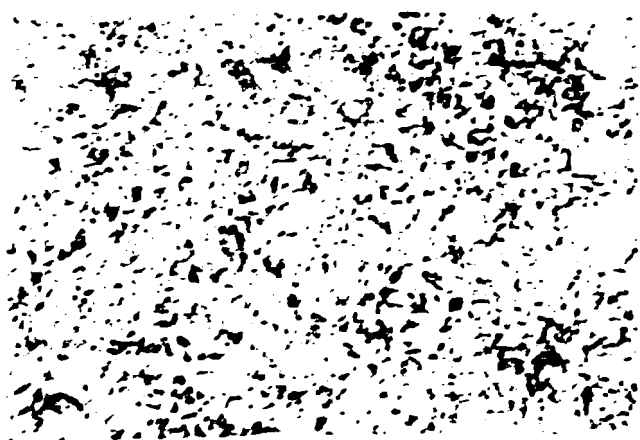


Fig.A.9.3.9.Oțel 0.3-0.4% C, recopt, mărire x500.

Microstructuri pentru oțel 0,42-0,5%C.

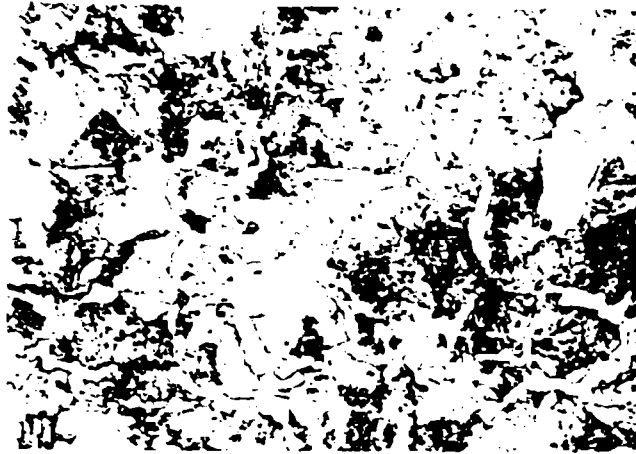


Fig.A.9.3.10.Oțel 0,42-0,5%C, laminat la cald, mărire x500.



Fig.A.9.3.11.Oțel 0,42-0,5%C, normalizat, mărire x500.

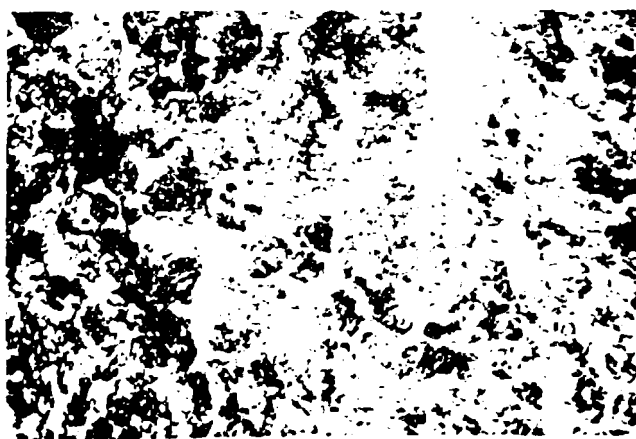


Fig.A.9.3.12.Oțel 0,42-0,5%C, recopt, mărire x500.

Tabel A.9.3.1

Tip oțel (%C)	Mărire	Stare material	Simbol utilizat în diagrame
Oțel. 0.08-0.15%C	x500	Laminat la cald	○
		Normalizat	×
		Recopt	+
Oțel. 0.16-0.25%C		Laminat la cald	○
		Normalizat	×
		Recopt	+
Oțel. 0.3-0.4%C		Laminat la cald	○
		Normalizat	×
		Recopt	+
Oțel. 0,42-0.5%C		Laminat la cald	○
		Normalizat	×
		Recopt	+

Diagrame rezultate în urma rulării programului „Identificare_B”

Oțel 0,08-0,15%C

$k_n = 1 \dots 255$, $i_{hmed} = 20$

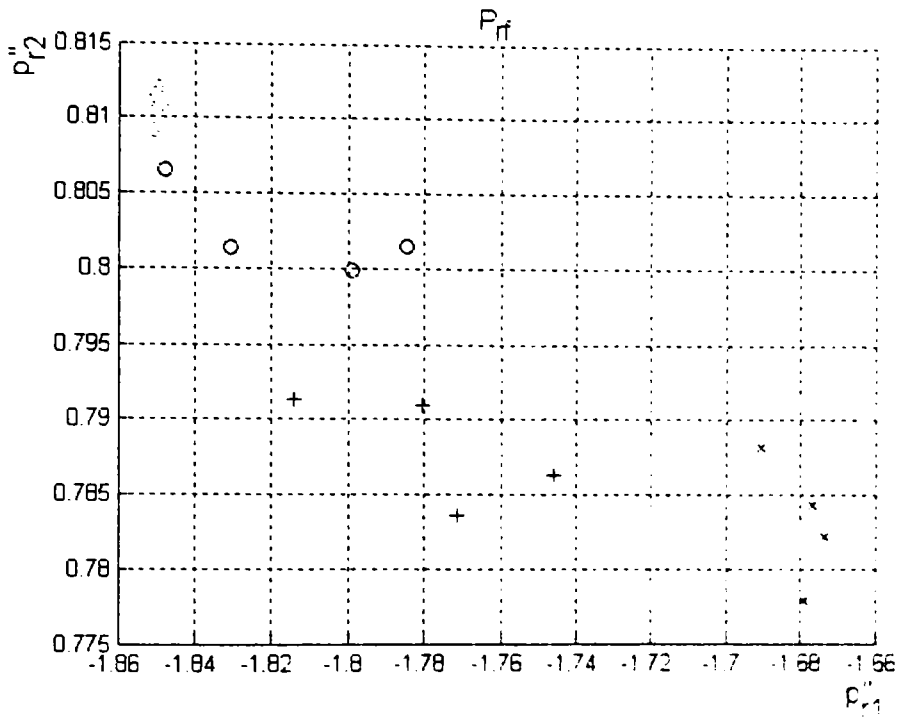


Fig.A.9.3.13 Diagrama P''_{r1} pentru oțel 0.08-0.15%C.

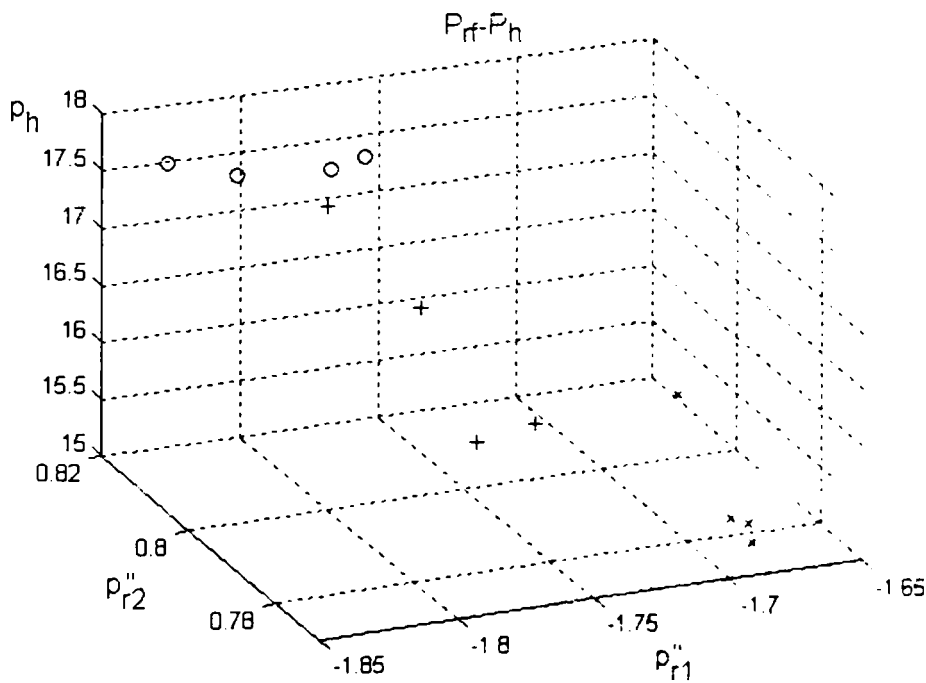


Fig.A.9.3.14 Diagrama P''_{r1} - P_h pentru oțel 0.08-0.15%C.

Oțel 0,16-0,25%C

$k_n = 1 \dots 255, i_{imed} = 20$

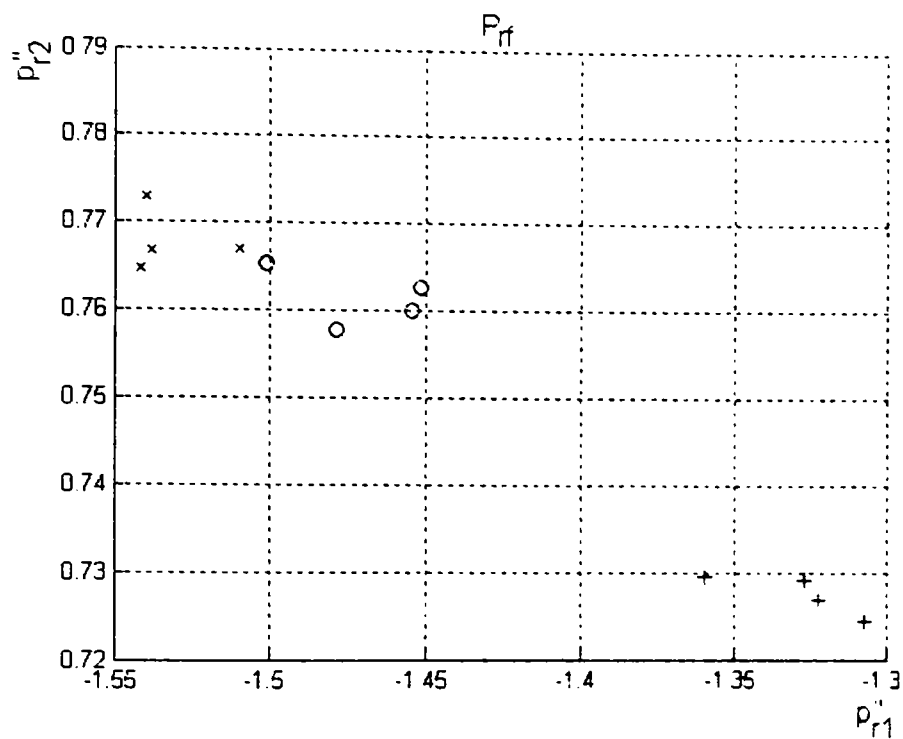


Fig.A.9.3.15 Diagrama P_{rf} pentru oțel 0,16-0,25%C.

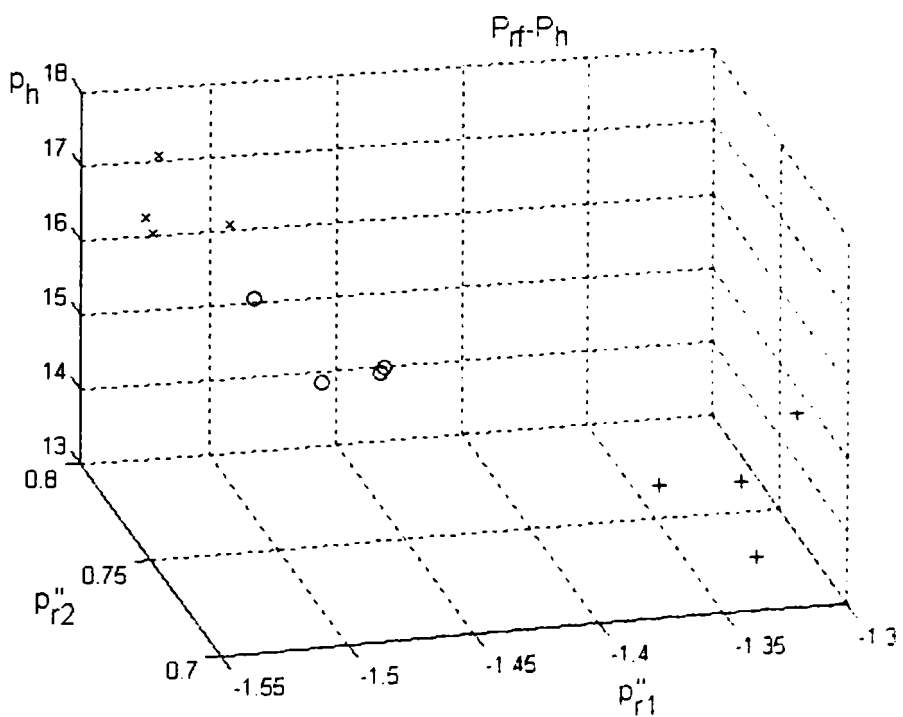


Fig.A.9.3.16 Diagrama $P_{rf}-P_h$ pentru oțel 0,16-0,25%C.

Oțel 0,3-0,4%C

$k_n = 1 \dots 255, i_{hmed} = 20$

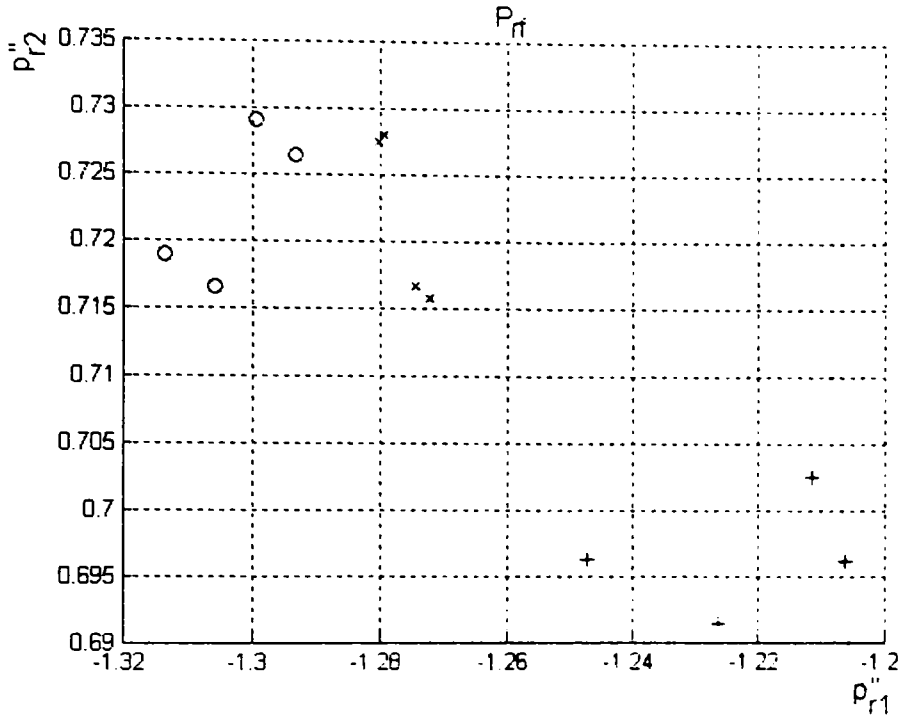


Fig.A.9.3.17 Diagrama P''_{r1} pentru oțel 0.3-0.4%C.

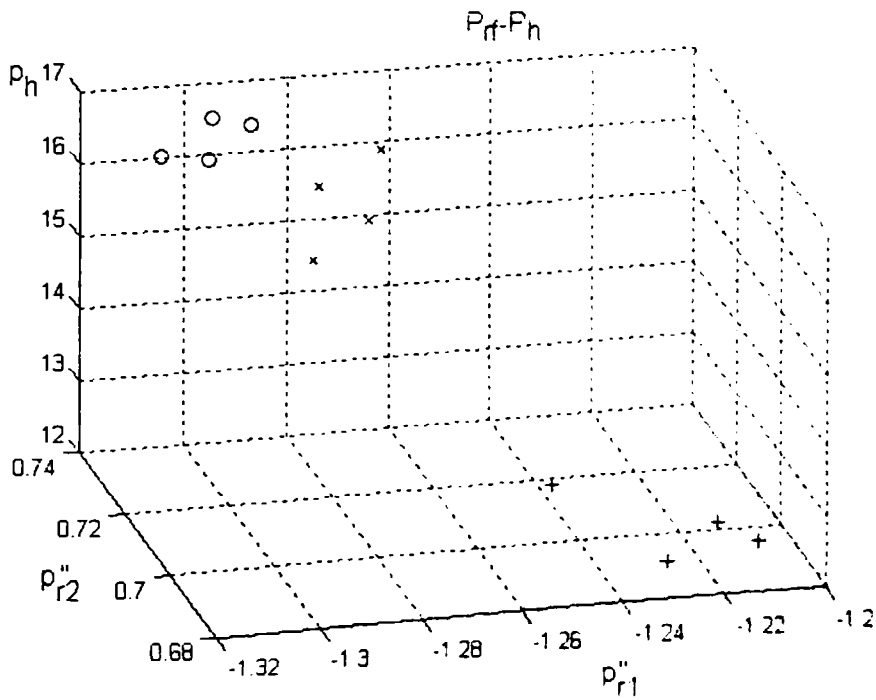


Fig.A.9.3.18. Diagrama P''_{r1} - P_h pentru oțel 0.3-0.4%C.

Oțel 0,42-0,5%C

$k_n = 1 \dots 255, i_{hmed} = 20$

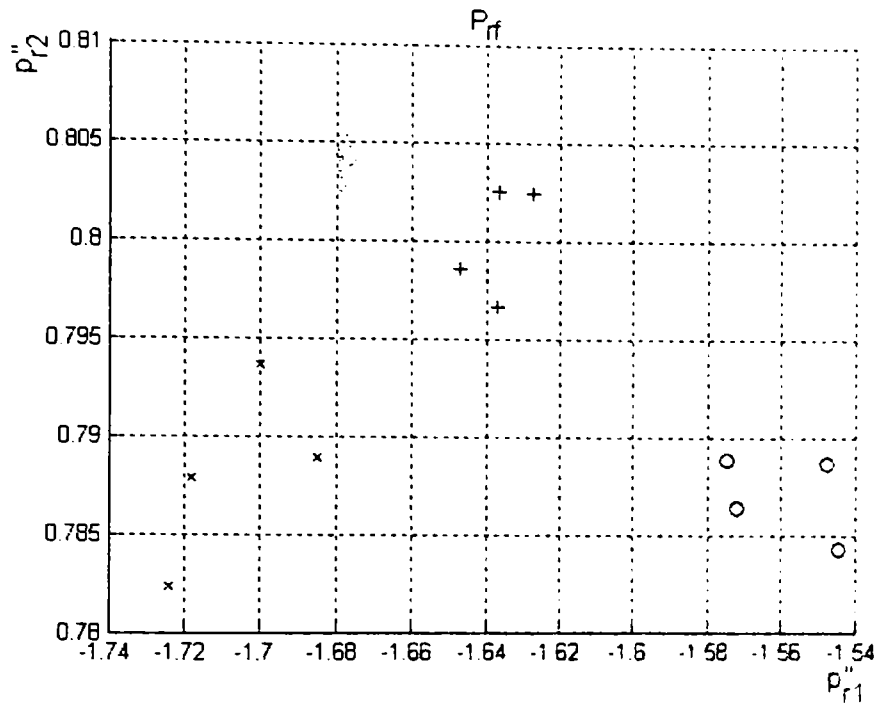


Fig.A.9.3.19 Diagrama P''_{r1} pentru oțel 0.42-0.5%C.

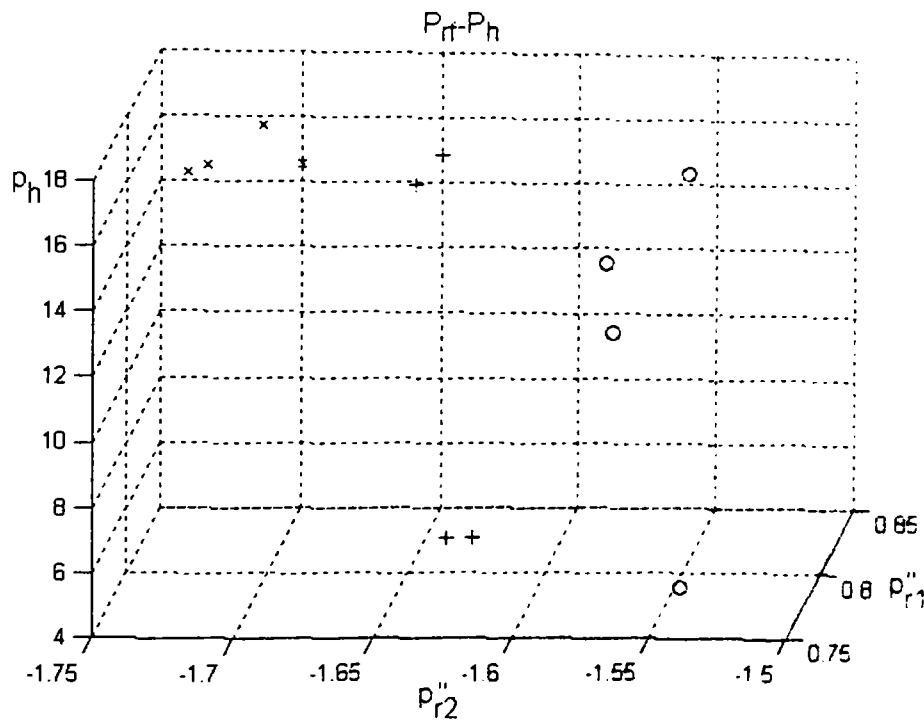


Fig.A.9.3.20 Diagrama $P''_{r1}-P_h$ pentru oțel 0.42-0.5%C.

**Program de identificare a microstructurilor în cadrul analizelor de tip
B: Identificare_B.**

```

global p1 ma p2 ma p1f p2f sigmaa
% 0.06 - 0.15%
datestr(now)
    k = 255;
    figure;
    nr_val1 = 1;
    nr_val2 = k;
    addpath('C:\Labcel\Tezafinal\final\Cap9\Recunoasterestructurii\Bazadeimagini\006');
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_laminat_la_cald_1.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'ko'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_laminat_la_cald_2.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'ko'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_laminat_la_cald_3.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'ko'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_laminat_la_cald_4.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'ko'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_normalizat_1.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'kx'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_normalizat_2.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'kx'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_normalizat_3.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'kx'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_normalizat_4.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'kx'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_recopt_1.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'k+'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_recopt_2.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'k+'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_recopt_3.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'k+'); hold on
    y0 = double(imread('Otel_0_06-0_15%C_nital2%_x500_recopt_4.bmp'));
Cal_fft2Dsuma;
    plot3(p1,p2,imh.'k+'); hold on
grid on;
datestr(now)

```

Subprogram de calcul a parametrilor fractali : Cal_fft2Dsuma

```
dimensx = (1:512);
x = reshape(repmat(dimensx,512,1),512*512,1);
dimensy = (1:512)';
y = repmat(dimensy,512,1);
x = x - 256;
y = y - 256;
Y = fft2(y0);
Z = fftshift(Y);
Pyy2 = double(uint8(log10((Z.*conj(Z))/(512*512)).^2)));
z = reshape(Pyy2,512*512,1);
r = round(sqrt(x.^2 + y.^2));
pcart = [r z];
pcart_r = sortrows(pcart,1);
lim = (find(pcart_r(:,1)<256));
lim2 = lim(end);
pcart_r = pcart_r(1:lim2,:);
for i = 1:255
    r_1 = find(pcart_r(:,1) == i);
    lim1_1 = r_1(1);
    lim1_2 = r_1(end);
    val_r_suma(i) = sum(pcart_r(lim1_1:lim1_2,2))/(lim1_2-lim1_1);
    x1(i) = i;
end
x_log = log10(x1);
val_r_suma_log = log10(val_r_suma);
p0 = polyfit((x_log(nr_val1:nr_val2)),(val_r_suma_log(nr_val1:nr_val2)),1);
f_val = polyval(p0,x1);
p1x = p0(1);
p2y = p0(2);
p1 = sqrt(2)/2*(p1x - p2y);
p2 = sqrt(2)/2*(p2y + p1x);
% Histograma
bins = 20;
[counts,x] = imhist(uint8(y0),bins);
[m imh] = max(counts);
```

Baza de imagini utilizată pentru analizelor de tipul C.

Microstructuri ferito – perlitice în șiruri, oțel până la 0,15%C, mărire x100.



Fig.A.9.5.1.Oțel până la 0.15%C. structură ferito-perlitică în șiruri, mărire x100, punctaj = 0.



Fig.A.9.5.2.Oțel până la 0.15%C. structură ferito-perlitică în șiruri, mărire x100, punctaj = 1.



Fig.A.9.5.3.Oțel până la 0.15%C. structură ferito-perlitică în șiruri, mărire x100, punctaj = 2.

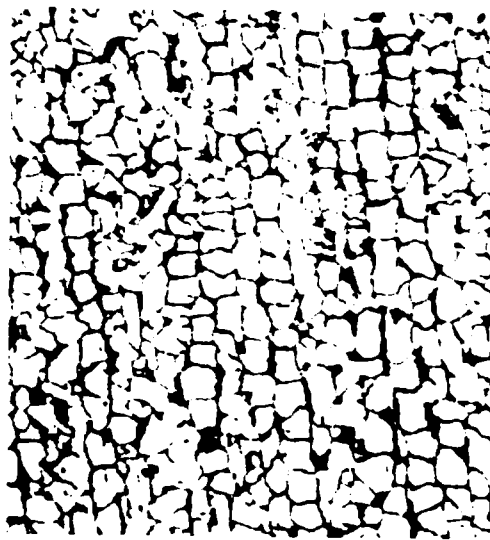


Fig.A.9.5.4.Oțel până la 0.15%C, structură ferito-perlitică în șiruri, mărire x100, punctaj = 3.

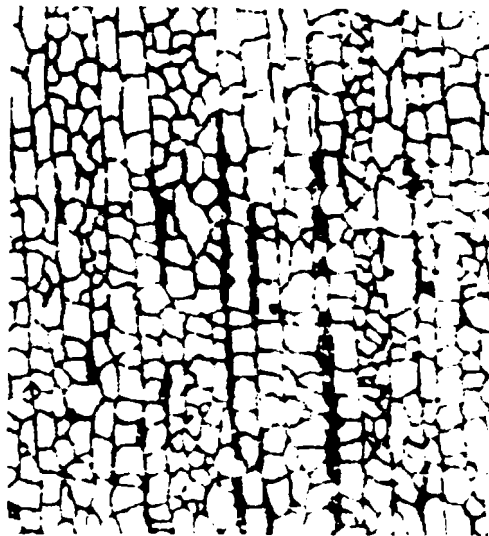


Fig.A.9.5.5.Oțel până la 0.15%C, structură ferito-perlitică în șiruri, mărire x100, punctaj = 4.

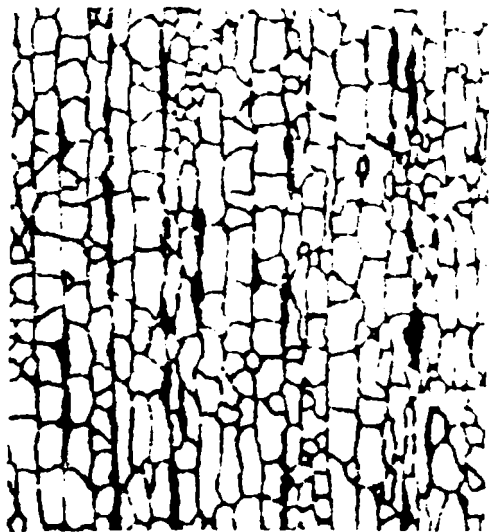


Fig.A.9.5.6.Oțel până la 0.15%C, structură ferito-perlitică în șiruri, mărire x100, punctaj = 5.



Fig.A.9.5.7.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 1.

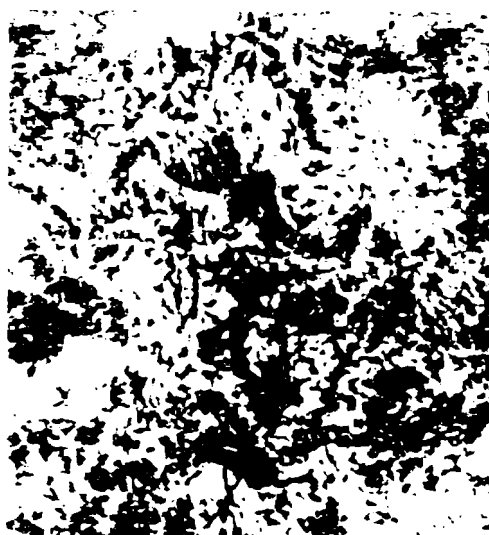


Fig.A.9.5.8.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 2.



Fig.A.9.5.9.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 3.



Fig.A.9.5.10.Oțel 0.15% C, determinare finete perlită, mărire x1000, punctaj = 4.



Fig.A.9.5.11.Oțel 0.15% C, determinare finete perlită, mărire x1000, punctaj = 5.



Fig.A.9.5.12.Oțel 0.15% C, determinare finete perlită, mărire x1000, punctaj = 6.

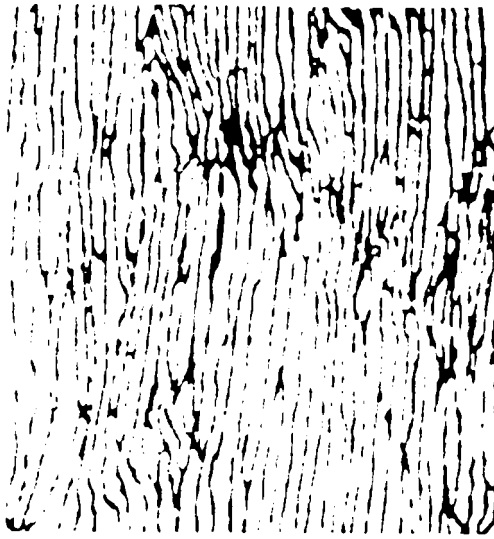


Fig.A.9.5.13.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 7.



Fig.A.9.5.14.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 8.



Fig.A.9.5.15.Oțel 0.15%C, determinare finețe perlită, mărire x1000, punctaj = 9.

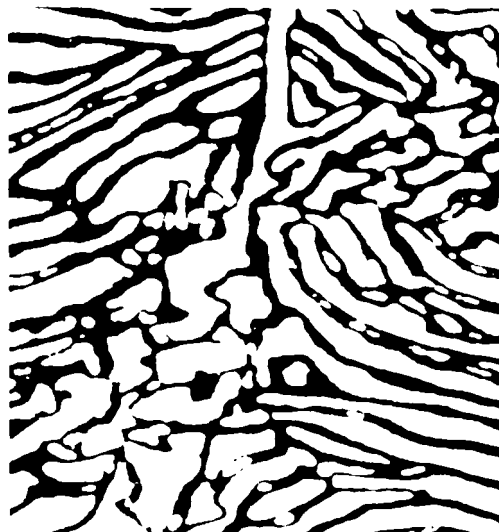


Fig.A.9.5.16. Oțel 0.15% C. determinare finete perlită. mărire x1000. punctaj - 10

Microstructuri, determinarea suprafeței perlitei la fonte cu grafit nodular, mărire x100.

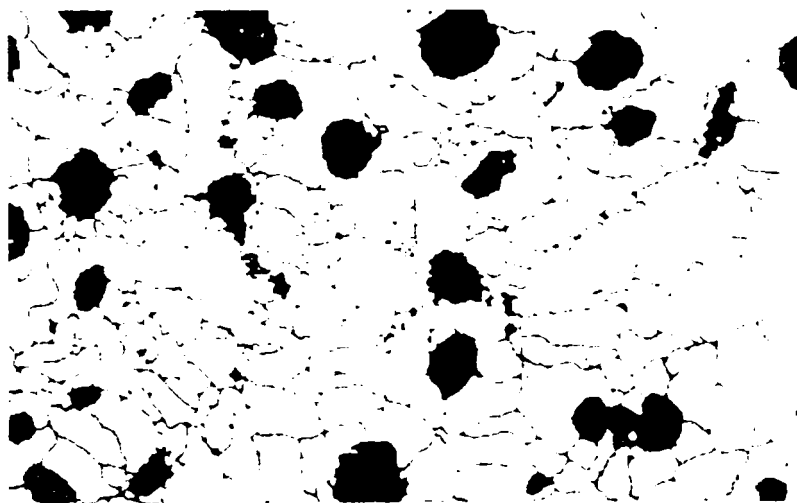


Fig.A.9.5.17. Microstructură cu suprafața perlitei de 2% în fontă cu grafit nodular. mărire x100.

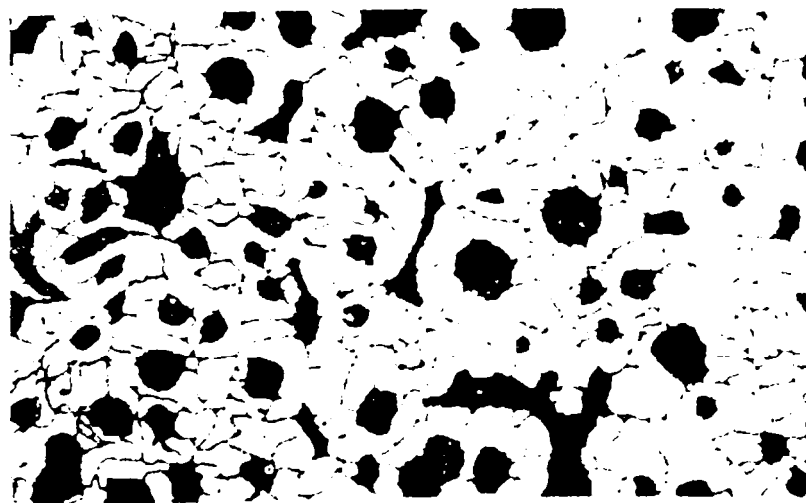


Fig.A.9.5.18. Microstructură cu suprafața perlitei între 2% și 10% în fontă cu grafit nodular. mărire x100.

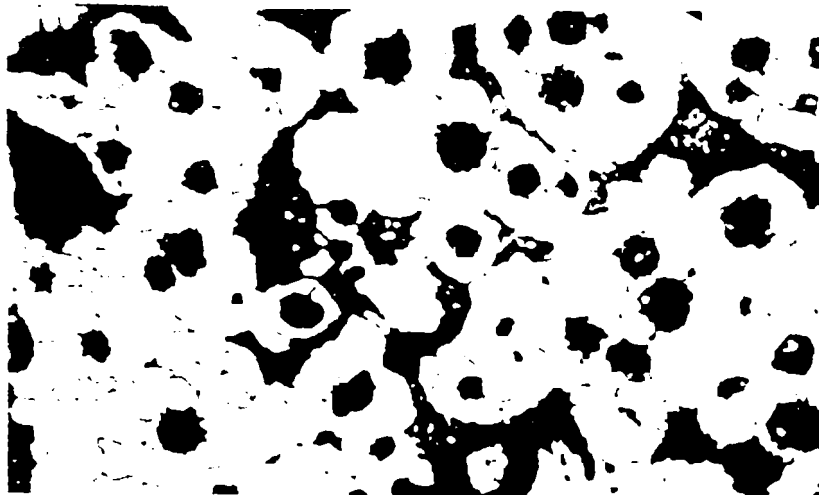


Fig.A.9.5.19. Microstructură cu suprafața perlitei între 10% și 40% în fontă cu grafit nodular, mărire x100.

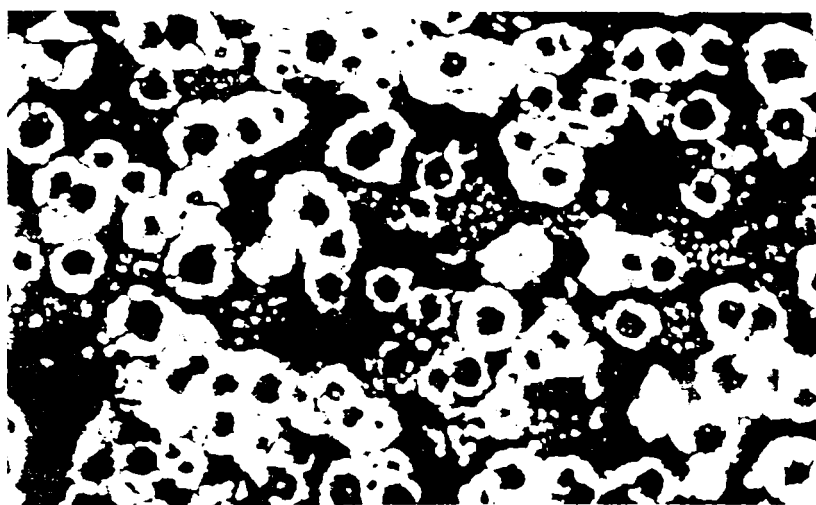


Fig.A.9.5.20. Microstructură cu suprafața perlitei între 40% și 65% în fontă cu grafit nodular, mărire x100.

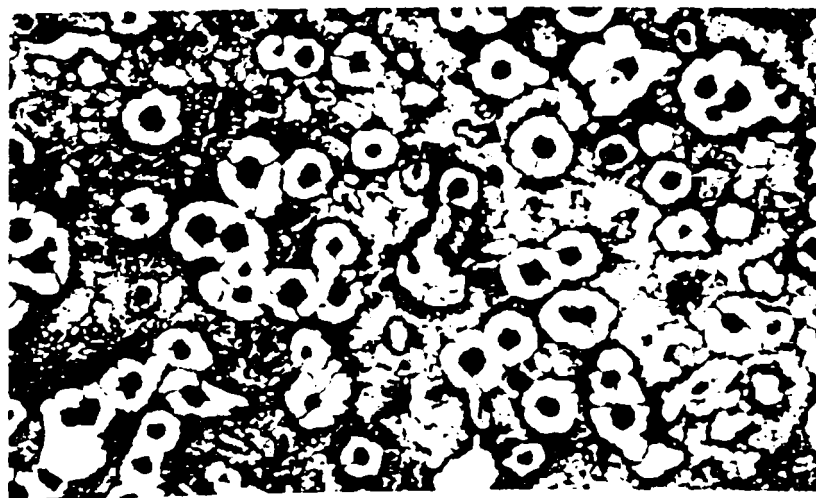


Fig.A.9.5.21. Microstructură cu suprafața perlitei între 65% și 80% în fontă cu grafit nodular, mărire x100.

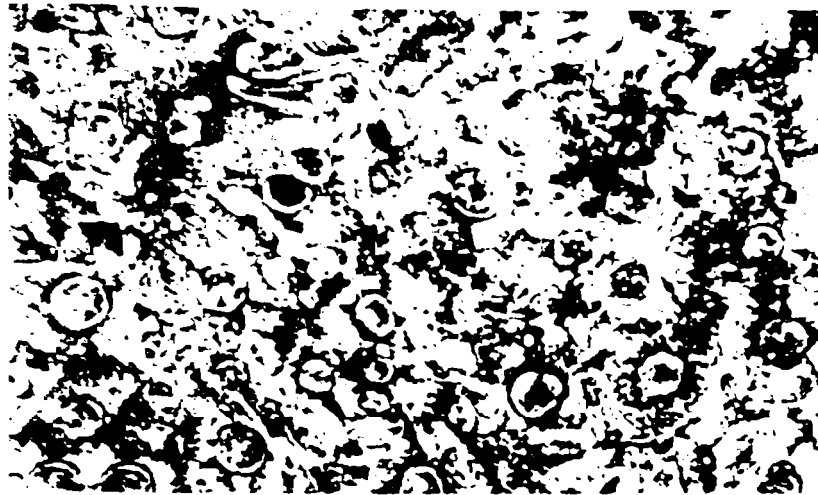


Fig.A.9.5.22. Microstructură cu suprafața perlitei de 80% în fontă cu grafit nodular.
mărire x100.

*Microstructuri, determinarea lungimii lamelelor la fonte cu grafit lamelar,
mărire x100.*

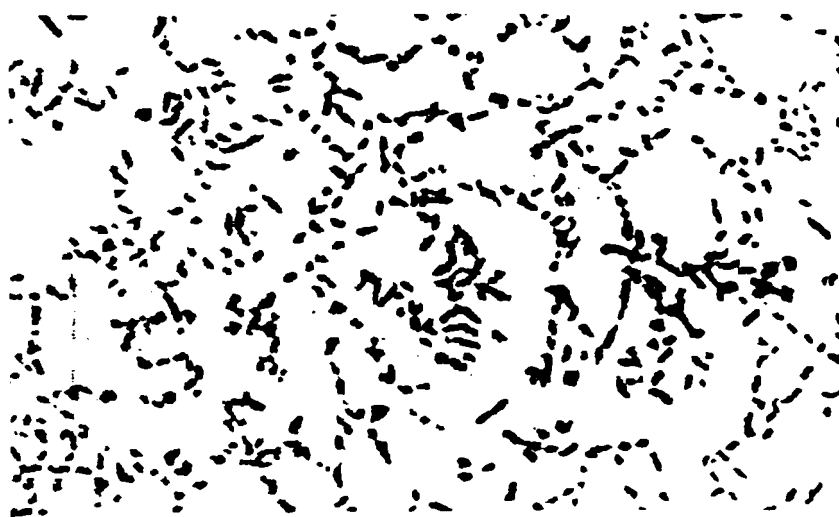


Fig.A.9.5.23. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 1, mărire x100.

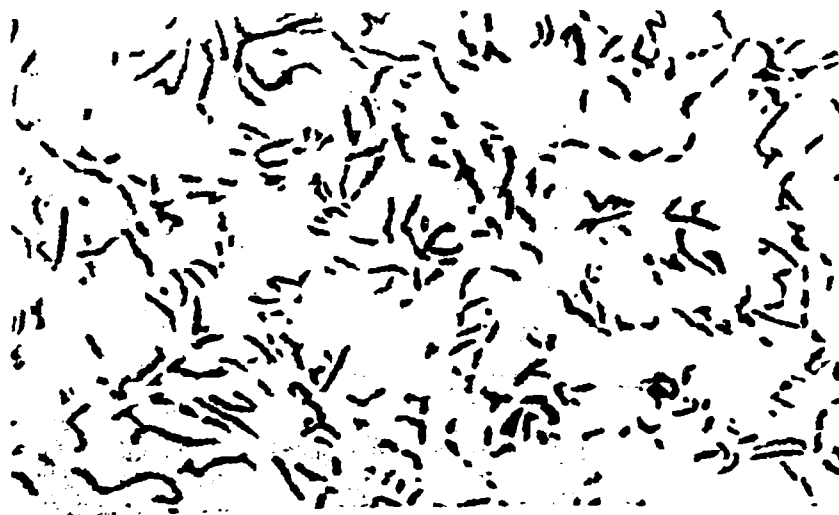


Fig.A.9.5.24. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 2, mărire x100.



Fig.A.9.5.25. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 3, mărire x100.



Fig.A.9.5.26. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 4, mărire x100.



Fig.A.9.5.27. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 5, mărire x100.



Fig.A.9.5.28. Microstructură cu lungimea lamelelor în fontă cu grafit lamelar.
punctaj = 6. mărire x100.

Tabel A.9.5.1

Tip analiză	Mărire	Punctaj sau % constituent	Simbol utilizat în diagrame
Determinarea punctajului, microstructuri ferito – perlitice în șiruri, oțel până la 0.15%C	x100	0	○
		1	×
		2	+
		3	✦
		4	□
		5	◇
Determinarea punctajului pentru fineței perlitice, oțel 0,15%C	x1000	1	○
		2	×
		3	+
		4	✦
		5	□
		6	◇
		7	▽
		8	△
		9	▷
		10	◁
Determinarea procentului suprafeței perlitice la fonte cu grafit nodular	x100	2%	○
		2% - 10%	×
		10% - 40%	+
		40% - 65%	✦
		65% - 80%	□
		80%	◇
Determinarea punctajului corespunzător lungimii lamelelor la fonte cu grafit lamelar	x100	1	○
		2	×
		3	+
		4	✦
		5	□
		6	◇

Program pentru determinarea punctajului ferito-perlitic: E1.

```

global p1 p2 p3
datestr(now)
addpath('C:\Label\Teza\final\final\Cap9\Recunoasterestructurii-Bazadeimagini Ferito_perlitica ');
figure;
y0 = double((imread('ferito_perlitica_0_15%C_x100_0_1.bmp')));
FFT2D;
p01 = p1;
plot(1,p1,'ko'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_0_2.bmp')));
FFT2D;
p02 = p1;
plot(1,p1,'ko'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_0_3.bmp')));
FFT2D;
p03 = p1;
plot(1,p1,'ko'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_0_4.bmp')));
FFT2D;
p04 = p1;
plot(1,p1,'ko'); hold on
fin(1) = mean([p01 p02 p03 p04]);
%%%%%%%%%%
y0 = double((imread('ferito_perlitica_0_15%C_x100_1_1.bmp')));
FFT2D;
p01 = p1;
plot(2,p1,'kx'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_1_2.bmp')));
FFT2D;
p02 = p1;
plot(2,p1,'kx'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_1_3.bmp')));
FFT2D;
p03 = p1;
plot(2,p1,'kx'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_1_4.bmp')));
FFT2D;
p04 = p1;
plot(2,p1,'kx'); hold on
fin(2) = mean([p01 p02 p03 p04]);
%%%%%%%%%%
y0 = double((imread('ferito_perlitica_0_15%C_x100_2_1.bmp')));
FFT2D;
p01 = p1;
plot(3,p1,'k+'); hold on
y0 = double((imread('ferito_perlitica_0_15%C_x100_2_2.bmp')));
FFT2D;
p02 = p1;

```

```

    plot(3,p1,'k+'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_2_3.bmp')));
FFT2D;
p03 = p1;
    plot(3,p1,'k+'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_2_4.bmp')));
FFT2D;
p04 = p1;
    plot(3,p1,'k+'); hold on
fin(3) = mean([p01 p02 p03 p04]);
%%%%%%%%%
    y0 = double((imread('ferito_perlitica_0_15%C_x100_3_1.bmp')));
FFT2D;
p01 = p1;
    plot(4,p1,'k*'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_3_2.bmp')));
FFT2D;
p02 = p1;
    plot(4,p1,'k*'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_3_3.bmp')));
FFT2D;
p03 = p1;
    plot(4,p1,'k*'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_3_4.bmp')));
FFT2D;
p04 = p1;
    plot(4,p1,'k*'); hold on
fin(4) = mean([p01 p02 p03 p04]);
%%%%%%%%%
    y0 = double((imread('ferito_perlitica_0_15%C_x100_4_1.bmp')));
FFT2D;
p01 = p1;
    plot(5,p1,'ks'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_4_2.bmp')));
FFT2D;
p02 = p1;
    plot(5,p1,'ks'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_4_3.bmp')));
FFT2D;
p03 = p1;
    plot(5,p1,'ks'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_4_4.bmp')));
FFT2D;
p04 = p1;
    plot(5,p1,'k'); hold on
fin(5) = mean([p01 p02 p03 p04]);
%%%%%%%%%
    y0 = double((imread('ferito_perlitica_0_15%C_x100_5_1.bmp')));
FFT2D;
p01 = p1;
    plot(6,p1,'kd'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_5_2.bmp')));

```



```

FFT2D:
p02 = p1;
    plot(6,p1,'kd'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_5_3.bmp')));
FFT2D:
p03 = p1;
    plot(6,p1,'kd'); hold on
    y0 = double((imread('ferito_perlitica_0_15%C_x100_5_4.bmp')));
FFT2D:
p04 = p1;
    plot(6,p1,'kd'); hold on
fin(6) = mean([p01 p02 p03 p04]);
grid on
% Aproximare
x = [1 2 3 4 5 6];
x1 = (1:0.1:6);
p00 = polyfit(x,fin.3);
f_val = polyval(p00,x1);
plot(x1,f_val,'k'); hold on
datestr(now)

```

Subprogram : E2

```

Y = fft2(y0);
Z = fftshift(Y);
Pyy2 = double(uint8(log10((Z.*conj(Z))/(512*512)).^2));
p1 = (sum(sum(Pyy2(248:264,1:64))));
% Histograma
bins = 50;
[counts.x] = imhist(uint8(y0),bins);
[m imh] = max(counts);
p3 = imh;
if p3>42.5
    hhh = 1;
end
if p3<=42.5
    hhh = 1.15;
end
p1 = p1*hhh/6000;

```

Program pentru stabilirea punctajului lungimii lamelelor din fontele cu grafit lamelar : LL1

```

global p1 p2 p3
% Lungime_lamele
addpath('C:\Labcel\Tezafinal\final\Cap9Recunoasterestructurii\Bazadeimagini Ferite_perlitica ');
figure:
y0 = double(imread('Lungime_lamele_grafit_1_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(1,p3,'ko'); hold on
    y0 = double(imread('Lungime_lamele_grafit_1_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(1,p3,'ko'); hold on
    y0 = double(imread('Lungime_lamele_grafit_1_x100_3.bmp'));
LL2:
z(3) = p3;
    plot(1,p3,'ko'); hold on
    y0 = double(imread('Lungime_lamele_grafit_1_x100_4.bmp'));
LL2:
z(4) = p3;
    plot(1,p3,'ko'); hold on
zz(1) = mean(z);
    y0 = double(imread('Lungime_lamele_grafit_2_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(2,p3,'kx'); hold on
    y0 = double(imread('Lungime_lamele_grafit_2_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(2,p3,'kx'); hold on
    y0 = double(imread('Lungime_lamele_grafit_2_x100_3.bmp'));
LL2:
z(3) = p3;
    plot(2,p3,'kx'); hold on
    y0 = double(imread('Lungime_lamele_grafit_2_x100_4.bmp'));
LL2:
z(4) = p3;
    plot(2,p3,'kx'); hold on
zz(2) = mean(z);
y0 = double(imread('Lungime_lamele_grafit_3_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(3,p3,'k+'); hold on
    y0 = double(imread('Lungime_lamele_grafit_3_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(3,p3,'k+'); hold on

```

```

    y0 = double(imread('Lungime_lamele_grafit_3_x100_3.bmp'));
LL2:
z(3) = p3;
    plot(3,p3,'k+'); hold on
    y0 = double(imread('Lungime_lamele_grafit_3_x100_4.bmp'));
LL2:
z(4) = p3;
    plot(3,p3,'k+'); hold on
zz(3) = mean(z);
y0 = double(imread('Lungime_lamele_grafit_4_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(4,p3,'k*'); hold on
    y0 = double(imread('Lungime_lamele_grafit_4_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(4,p3,'k*'); hold on
    y0 = double(imread('Lungime_lamele_grafit_4_x100_3.bmp'));
LL2:
z(3) = p3;
    plot(4,p3,'k*'); hold on
    y0 = double(imread('Lungime_lamele_grafit_4_x100_4.bmp'));
LL2:
z(4) = p3;
    plot(4,p3,'k*'); hold on
zz(4) = mean(z);
y0 = double(imread('Lungime_lamele_grafit_5_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(5,p3,'ks'); hold on
    y0 = double(imread('Lungime_lamele_grafit_5_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(5,p3,'ks'); hold on
    y0 = double(imread('Lungime_lamele_grafit_5_x100_3.bmp'));
LL2:
z(3) = p3;
    plot(5,p3,'ks'); hold on
    y0 = double(imread('Lungime_lamele_grafit_5_x100_4.bmp'));
LL2:
z(4) = p3;
    plot(5,p3,'ks'); hold on
zz(5) = mean(z);
y0 = double(imread('Lungime_lamele_grafit_6_x100_1.bmp'));
LL2:
z(1) = p3;
    plot(6,p3,'kd'); hold on
    y0 = double(imread('Lungime_lamele_grafit_6_x100_2.bmp'));
LL2:
z(2) = p3;
    plot(6,p3,'kd'); hold on
    y0 = double(imread('Lungime_lamele_grafit_6_x100_3.bmp'));

```

```

LL2:
z(3) = p3:
    plot(6,p3,'kd'): hold on
    y0 = double(imread('Lungime_lamele_grafit_6_x100_4.bmp'));
LL2:
z(4) = p3:
    plot(6,p3,'kd'): hold on
zz(6) = mean(z):
grid on
x = [1 2 3 4 5 6];
x1 = (1:0.1:6):
ftype = fitype('a*exp(b*x+c)');
[fit2.gof2.out2] = fit(x'.zz'.ftype);
plot(x1,fit2.a*exp(fit2.b*x1+fit2.c),'k')

```

Subprogram LL2

```

I = medfilt2(y0,[5 5]);
BW2 = im2bw(uint8(I), graythresh(uint8(I)));
BW = imcomplement(BW2);
[L num]= bwlabel(BW);
RGB = label2rgb(L);
stats = regionprops(L, 'MajorAxisLength');
mal = [stats.MajorAxisLength];
p3 = (mean(abs(mal)./num));

```