

43/1980

L. A. Węgrowicz, J. Radzki

PODWÓJNA ANTENA V
JAKO ŹRÓDŁO PIERWOTNE
DLA GŁĘBOKICH ZWIERCIADEŁ
PARABOLICZNYCH

Część II

P. 269



WARSZAWA 1980

<http://rcin.org.pl>

Praca wpłynęła do Redakcji dnia 29 października 1980 r.

Zarejestrowana pod nr 43/1980



57118



Na prawach rękopisu

Instytut Podstawowych Problemów Techniki PAN
Nakład 160 egz. Ark.wyd. 2,4. Ark.druk. 4,25
Oddano do drukarni w listopadzie 1980 r.
Nr zamówienia 734/0

Warszawska Drukarnia Naukowa, Warszawa,
ul.Sniadeckich 8

<http://rcin.org.pl>

Lucjan A. Węgrowicz
Inst. Podstawowych Problemów Techniki PAN,
Obserwatorium Astronomiczne UJ

Jerzy Radzki
Inst. Podstaw Informatyki PAN

PODWÓJNA ANTENA V
JAKO ŹRÓDŁO PIERWOTNE
DLA GŁĘBOKICH ZWIERCIADEŁ PARABOLICZNYCH
Część II
Wyznaczanie pola w aperturze zwierciadła
program „APERTURE”

Streszczenie

W pracy tej przedstawione zostały rezultaty rozważań na temat wykorzystania anteny V jako elementu źródła pierwotnego do oświetlania głębokich zwierciadeł parabolicznych. Część druga poświęcona jest wyznaczaniu parametrów pola składowej podstawowej i crosspolaryzacyjnej, a także elipsy polaryzacyjnej, w aperturze zwierciadła oświetlonego podwójną anteną V. Obliczenia te realizuje numerycznie opracowany i przytoczony program „APERTURE”. Wyniki obliczeń dostosowane są do urządzeń peryferyjnych typu „plotter” lub „display” i pozwalają przygotowywać „mapy synoptyczne” pola w aperturze zwierciadła.

Wstęp

Na podstawie znajomości rozkładu pola składowej podstawowej i crosspolaryzacyjnej w aperturze wysnuć można istotne dla dalszej analizy wnioski o efektywności oświetlenia zwierciadła. Wnioski te są szczególnie cenne w pośrednich

stadiach analizy, zanim jeszcze wykonane zostały czasochłonne obliczenia parametrów całkowitych zwierciadła, takich jak zysk, funkcje kierunkowe obu składowych oraz sprawność oświetlenia.

Rozkład pola w aperturze anteny wygodnie jest opisywać za pomocą „map synoptycznych” tj. rysunków powierzchni apertury z naniesionymi izoliniami natężenia pola i izoliniami parametrów elipsy polaryzacyjnej. Znajomość „map synoptycznych” rozkładu pola na aperturze pozwala zmniejszyć objętość dalszych czasochłonnych obliczeń, a także głębiej wniknąć w proces formowania pola promieniowania. Sprzyja ona także powstawaniu właściwych pojęć intuicyjnych.

Część II niniejszej pracy poświęcona jest omówieniu zaproponowanej metody przygotowywania „map synoptycznych” rozkładu pola w aperturze zwierciadła i zamyka się realizującym tą metodę numerycznie dwuczęściowym programem APERTURE . W programie tym wykorzystano szereg procedur przygotowanych i wytestowanych dla programu SOURCE III**, których opis zamieszczony został w części I. W części drugiej omówione zostaną tylko te procedury, które, bądź wogóle nie występują w programie SOURCE III, bądź też zostały zmodyfikowane.

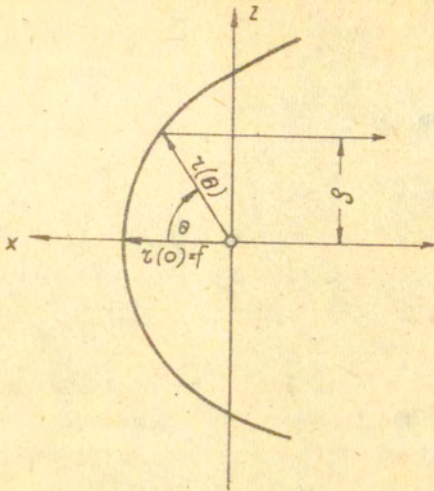
Układy współrzędnych na aperturze

Procedury TRANS 4 i TRANS 5

Oprócz omówionego w części I układu współrzędnych prostokątnych (y, z) , wygodnie jest dysponować na aperturze układem współrzędnych biegunowych (x, φ) , które lepiej niż współrzędne prostokątne wpisują się w aperturę kołową. Dla określenia pod jakimi kątami (ξ, θ) we współrzędnych sferycznych widziany jest z ogniska punkt o współrzędnych (x, φ) na aperturze służy procedura TRANS 4(chi, ro, ksi, th).

Przy opracowywaniu procedury skorzystano z zależności (I.31), z której dla dowolnego ξ wynika

$$(1) \quad r(\theta) = \frac{2f}{1 + \cos(\theta)}$$



Rys.1

Posługując się rysunkiem 1. nietrudno ustalić, że

$$(2) \quad \xi(\theta) = r(\theta) \sin \theta$$

a następnie

$$(3) \quad \frac{\xi(\theta)}{f} = \frac{2 \sin(\theta)}{1 + \cos(\theta)}$$

skąd ostatecznie

$$(4) \quad \theta = \arccos \frac{4 - \left(\frac{\xi}{f}\right)^2}{4 + \left(\frac{\xi}{f}\right)^2}$$

ponieważ (4) zachodzi dla dowolnego ξ jest oczywiste, że

$$(5) \quad \xi = \chi$$

Zależności (4) i (5) wykorzystane zostały w procedurze.

Dla obliczania współrzędnych prostokątnych (y, z) punktu o współrzędnych biegunowych (χ, ρ) korzysta się z zależności

$$(6) \quad z = \rho \cos(\chi)$$

$$(7) \quad y = \rho \sin(\chi)$$

które ujęte zostały w procedurę TRANS 5(χ, ρ, y, z).

Wyszukiwanie zadanych wartości funkcji
Procedury CARD i AUXCARD

Przygotowanie map synoptycznych funkcji opisujących zachowanie się pola elektromagnetycznego na aperturze zwierciadła polega na odszukaniu ciągów par współrzędnych y_1 i z_1 spełniających równania postaci

$$(8) \quad f(y_1, z_1) = c_j$$

Ze względu na złożoną strukturę występujących u nas funkcji f , obliczanie których wymaga stosowania skomplikowanych procedur PRIMESOURCE, MNOZ SIAT i MN REFL oraz łańcucha procedur TRANS zrezygnowano z różniczkowania f i bezpośredniego znajdowania y_1 i z_1 na rzecz zastosowania trzypunktowej metody interpolacyjnej realizowanej przez procedurę CARD($y, zp, dz, jz, izo, undef, discont, elips, F, Y, Z$). Procedura korzysta z zapamiętanych w tablicy F wartości f obliczonych w węzłach równoodległej siatki punktów.

Zadane z monitora wartości początkowe yp, zp , kroki dy i dz oraz wartości końcowe yk i zk współrzędnych siatki badane są przez procedurę AUXCARD($zp, dz, zk, jz, yp, dy, yk, iy$), która ustala nowe zp, zk, yp, yk tak by:

- a) punkty o współrzędnych $y=0$ lub $z=0$ były zawsze punktami środkowymi odpowiednich trójek punktów wykorzystywanych w procesie interpolacyjnym,

- b) ilość jz punktów po z była zawsze liczbą nieparzystą, co zapewnia przygotowanie całkowitej ilości trójek wartości funkcji poddawanej procesowi interpolacji. Podobnie, dla ujednoczenia programu ustala się ilość iy punktów po y ,
- c) ostatnie punkty końcowych trójek wypadają zawsze w zk .

Przy ustalonym y , dla trzech kolejnych punktów z przez wartości funkcji $f(y,z)$ prowadzi się wielomian interpolacyjny

$$(9) \quad P_2(z) = pz^2 + qz + r$$

Wielomian ten przyrównuje się następnie do zadanego izo i znajduje się pierwiastki równania

$$(10) \quad P_2(z) = izo_j$$

Jeżeli pierwiastki wielomianu są rzeczywiste i leżą pomiędzy skrajnymi z rozpatrywanych aktualnie trzech punktów to odpowiednie wartości y_i oraz z_i będące współrzędnymi punktów j -tej izolinii $f(y,z)$ zapamiętuje się w tablicach Y i Z zliczając jednocześnie ich ilość na zmiennej nielokalnej il .

Współczynniki p, q i r znajduje się z układu trzech równań liniowych z wyznacznikiem Vandermonde'a równym $-2(dz)^3$.

Procedura rozróżnia i dostosowuje się do trzech możliwych przypadków: gdy funkcja f jest istotnie nieliniowa ($p \neq 0$) rozpatruje wielomian drugiego stopnia, gdy funkcja f jest prawie liniowa ($p \approx 0$) rozpatruje wielomian pierwszego stopnia, gdy funkcja f jest prawie stała ($p \approx 0$ oraz $q \approx 0$) bada czy $r \approx izo_j$. W tym ostatnim przypadku drukowana jest tylko informacja „ y ; od z_1 do z_3 ”.

Trzy parametry bulowskie undef, discont, elips pozwalają stosować procedurę CARD do funkcji nieparzystych lub nieokreślonych na osiach współrzędnych oraz do funkcji pozornie nieciągłych w punkcie jak np. argument liczby zespolonej

w otoczeniu 0 lub 2π . Wywołanie procedury CARD od undef = true powoduje omińnięcie trójki punktów, punkt środkowy której leży na Oy lub Oz. Wywołanie procedury CARD od discont = true powoduje w zależności od aktualnej wartości izo usunięcie nieciągłości pozornych przez dodanie lub odjęcie 360° od wartości f naruszających pozornie jej ciągłość. Jeżeli nieciągłość jest istotna i nie daje się usunąć, to odpowiednia trójka punktów zostaje pominięta.

Procedurę CARD od discon = true i elips = true wywołuje się przy interpolowaniu kąta nachylenia α elipsy polaryzacyjnej, wartości którego leżą w przedziale $(-90^\circ, +90^\circ)$. Następuje wówczas przesunięcie obliczonych wartości α i zadanego izo przez dodanie do nich 90° oraz usunięcie nieciągłości pozornych przez odjęcie lub dodanie 180° . Wyłączane są również trójki zawierające punkty z niekreślonymi α ($\alpha = 360$).

Obliczanie parametrów elipsy polaryzacyjnej

Procedura POLELIPS 1

Ze względu na niesymetryczny charakter zależności podstawowych wykorzystywanych w procedurze POLELIPS (cz.I, 42 i 43), procedurę tą opracowano ponownie korzystając z zaczerpniętych z [1] zależności

$$(11) \quad m = \sqrt{\frac{1}{2}(-g + \sqrt{g^2 - 4})}$$

gdzie

$$(12) \quad g = \frac{1}{\sin^2 \theta} \left(\frac{E_g^2}{E_d^2} + 2 \cos^2 \theta + \frac{E_d^2}{E_g^2} \right)$$

$$(13) \quad \theta = \theta_g - \theta_d$$

oraz

$$(14) \quad \alpha = \frac{1}{2} \arctg \left(\frac{2E_g E_d \cos \theta}{E_g^2 - E_d^2} \right)$$

i nazwano POLELIPS 1($E_g, E_d, \theta_g, \theta_d, m, \alpha, f, \text{ster}$). Ewentualne zastąpienie procedury POLELIPS procedurą POLELIPS 1 w programie SOURCE III** wymaga tylko wymiany ciała procedury i nagłówków.

Wykreślanie map synoptycznych

Procedura DRAW

Procedura AXES

Procedura OPIS

Procedura ZNAKI

Wykreślanie map synoptycznych funkcji opisujących zachowanie się pola na aperturze zwierciadła dokonywane jest przez urządzenie samopiszące. Urządzenie to korzysta z danych postaci

(15) Q
 X -012 Y -012 f1

 N

zapisanych na taśmie perforowanej. Q powoduje rozpoczęcie, a N zakończenie pracy pisaka, X i Y są współrzędnymi kolejnych punktów, a f równe 1 lub 2 steruje szybkością przesuwu pisaka. Wartości współrzędnych wyrażane są przez liczby całkowite z przedziału (-999, +999). Przygotowanie taśmy realizowane jest przez procedurę DRAW(X,Y,n,cx,cy,tx,ty), korzystającą z n zapamiętanych w tablicach X i Y par wartości współrzędnych punktów podlegających naniesieniu na wykres. Wartości c i t dobierane są tak by dokonywane w procedurze przekształcenie

(16) $\bar{x} = cx + t$

transformowało przedziały (x_{min}, x_{max}) i (y_{min}, y_{max}) w przedział (-990,+990) lub inny aktualnie wymagany. Poza tym procedura powoduje wyperforowanie na taśmie polecenia zaznaczenia punktu (t_x, t_y) oraz polecenia odprowadzenia pisaka do punktu (999,999) po wyczerpaniu tablic X, Y. Dla zabezpieczenia mechanizmu urządzenia przed uszkodzeniem, wartości \bar{x} i \bar{y} leżące poza przedziałem (-999,+999) zastępowane są wartościami na brzegu.

Procedura DRAW występuje w programie samodzielnie bądź w procedurze OPIS(izo,s,n,Y,Z), która powoduje drukowanie na monitorze informacji o aktualnie perforowanej izolacji, potrzebnej do opisywania taśmy.

Bezparametrowa procedura-ZNAKI powoduje wyperforowanie na taśmie łatwo zauważalnych znaków nie przyjmowanych przez urządzenie samopiszzące, które w wyraźny sposób oddzielają kolejne rodziny rysowanych izolacji.

Procedura AXES(Y,Z,n) oblicza wewnątrz zadanego prostokąta (yp,yk,zp,zk) współrzędne y i z punktów biegunowego układu współrzędnych, które zapamiętane w tablicach Y i Z wykorzystywane są następnie w procedurze DRAW do wykreślania układu współrzędnych.

Współpraca z bębniami

Procedura DP

Procedury TODR, FROMDR

Procedury ITODR, IFROMDR

Ze względu na małą pojemność pamięci operacyjnej emc ODRA 1204 i dużą ilość wyników pośrednich, które należy zapamiętać, są one umieszczane na bębnie. Do współpracy z bębniem służą procedury: DP, TODR, FROMDR, ITODR, IFROMDR.

Pierwsza z nich jest procedurą bezparametrową, która ustala adresy (drumplace) pierwszych elementów tablic umieszczanych na bębnie. Informacje o tablicach, ich adresach i zawartości podane są w tablicy 1.

Trójparametrowe procedury TODR (dp,n,A) i FROMDR (dp,n,A) służą odpowiednio do przesyłania na bęben i ściągania z bębna n-elementowych tablic A typu real, pierwszy element których znajduje się pod adresem dp. ✓

Dla przesyłania, bądź ściągania, tablic typu integer służą analogiczne procedury ITODR i IFROMDR.

Tablica 1

zawartość tablicy	nazwa	adres
moduły E_x	MX	0
moduły E_y	MY	10000
moduły E_z	MZ	20000
argumenty E_x	AX	30000
argumenty E_y	AY	40000
argumenty E_z	AZ	50000
kąty ϕ_i	FI	60000
kąty ksi	KSI	70000
kąty delta	DELTA	80000
kąty theta	TH	90000
wsp. eliptyczności	M	100000
kąty nachylenia osi elipsy	ALFA	110000
parametry ster	STER	120000
dane początkowe	V	130000
ilości punktów siatki	JZIIY	130050

Omówienie programu

Program napisany jest w języku ALGOL 1204 dla emc ODRA 1204. Duża objętość algorytmu wywołała konieczność podzielenia go na dwie części : APERTURE** cz.I i APERTURE** cz.II.

W programie zawarte i wykorzystane są 23 procedury, nazwy i wykorzystanie których podane są w tablicy 2.

Każda procedura opatrzona jest odpowiednim komentarzem. Część pierwsza programu oblicza i przesyła na bęben dane pośrednie, którymi są wymienione w tablicy 1. parametry składowych pola i elipsy polaryzacyjnej obliczone w węzłach równoodległej siatki punktów na aperturze, a także dane początkowe. Jednocześnie zapamiętywane są wartości maksymalne modułów składowych natężenia pola, które wykorzystuje się dla normalizacji wielkości drukowanych w cz.I i danych wejściowych dla procedury interpolacyjnej CARD w części II programu.

Tablica 2

lp.	nazwa	cz.I	cz.II.
1	COMPLEX	✕	
2	TRANS 1	✕	
3	TRANS 3	✕	
4	TRANS 4	✕	
5	TRANS 5		✕
6	FAZA ZERO	✕	
7	MNOZ SIAT	✕	
8	MN REFL	✕	
9	PRZEDZIAŁ	✕	
10	PRIMESOURCE	✕	
11	POLELIPS 1	✕	
12	TODRAM	✕	
13	FROMDRAM	✕	✕
14	ITODRAM	✕	
15	IFROMDRAM	✕	✕
16	DP	✕	
17	STERO	✕	
18	AUXCARD	✕	
19	AXES		✕
20	CARD		✕
21	DRAW		✕
22	OPIS		✕
23	ZNAKI		✕

W części drugiej programu, przygotowane uprzednio i ściągnięte z bębna dane po uprzednim znormalizowaniu wprowadzane są do procedury interpolacyjnej CARD, która wyszukuje, drukuje na drukarce i zapamiętuje współrzędne aktualnie liczonej izolacji. Zapamiętane współrzędne są następnie wykorzystywane przez procedurę DRAW, która przygotowuje taśmę perforowaną z danymi dla plottera. W odpowiednich momentach ukazuje się na monitorze informacja o tym, która izolacja jest aktualnie liczona. Na podstawie tej informacji można ręcznie opisywać taśmę.

Obliczenia wykonane przez część drugą można powtarzać korzystając z tych samych zapamiętanych na bębnie wyników części pierwszej, o ile zachodzi potrzeba doliczenia dodatkowych izolinii.

Dane

Dane do programu wprowadza się z monitora po odpowiednich komunikatach. Wszystkie wprowadzone dane drukowane są następnie na początku wydruku co pozwala na łatwą identyfikację wyników. Po komunikacie "string" zarówno w części pierwszej jak i drugiej wypisać należy w nawiasie stringowym dowolny tekst, np. datę i komentarz do aktualnych obliczeń. Wypisany tekst pojawi się w nagłówku wydruku i wyperforowany zostanie na początku taśmy dla plottera.

Poniżej przytoczony jest przykładowy tabulogram z monitora. Wielkości wprowadzone zaznaczono kółkiem.

Ponieważ procedura AUXCARD zmienia w ogólnym przypadku zadane wartości y_p, z_p, y_k i z_k , program w części pierwszej modyfikuje także kąt rozwarcia i promień apertury tak by zmodyfikowany promień apertury stał się równy $\max(y_p, z_p, y_k, z_k)$. Zmodyfikowane wartości kąta rozwarcia i zmodyfikowanego promienia apertury drukowane są w wynikach po danych wejściowych.

t
r^{ad}r 1 WAIT

AP⁴RTURA z dn 19.03.1980 cz I

n 5096 r5915

w13

t^uksts 6.10.80.
kat rozw di^ola (sto^oni^o) = r135.0
.5 kata rozw a^ort (sto^oni^o) = r110.0,
dlugosc rami^onia (lambda) = r.25,
odl^ogosc di^oli (lambda) = r.25,
r^zs. fazow^o di^oli (alfa/kc) = r1.0,
y od r-3.0, co r.05, do r0.0,
z od r-3.0, co r.2, do r0.0,

wait k17-maxmaxmy wczytan^o. k21-izotr, k22-ni^o tabl

END

t
A.ertura z dn 19.03.1980 cz II

3785 r7226
w13 s 6.10.80.

wait k16-osic, k18-nlot, k19-cala a,ert, k20-konic izo

wait czy k21, k22-nic tabl

wait o, isac tasm

wait o, isac tasm : osic
izolini my db: izo od r30.0, co r2.5, do r0.0,

wait czy k20

my = -30.00
my = -27.50
my = -25.00
my = -22.50
my = -20.00
my = -17.50
my = -15.00
my = -12.50
my = -10.00
my = -7.50
my = -5.00
my = -2.50

izolini argy (stonic): izo od r0.0, co r15.0, do r360.0,

wait czy k20

argy = .00
argy = 15.00
argy = 30.00
argy = 45.00
argy = 60.00
argy = 75.00
argy = 90.00
argy = 105.00
argy = 120.00
argy = 135.00
argy = 150.00
argy = 165.00

izolini mz db: izo od r30.0, co r2.5, do r10.0,

wait czy k20

mz = -30.00
mz = -27.50
mz = -25.00
mz = -22.50
mz = -20.00
mz = -17.50
mz = -15.00

izolini argz (stonic): izo od r0.0, co r15.0, do r360.0,

wait czy k20
argz = .00
argz= 165.00
argz= 180.00
argz= 345.00
argz= 360.00

izolini^e m izo od 10.1, co 10.1, do 10.9,

wait czy k20

m= .10

m= .20

m= .30

m= .40

m= .50

m= .60

m= .70

m= .80

izolini^e alfa (sto_{ni}^e): izo od -90.0, co 15.0, do 90.0,

wait czy k20

alfa= -90.00

alfa= -75.00

alfa= -60.00

alfa= -45.00

alfa= -30.00

alfa= -15.00

alfa= .00

alfa= 15.00

alfa= 30.00

alfa= 45.00

alfa= 60.00

alfa= 75.00

alfa= 90.00

—

Sterowanie programem

Klucze

W obu częściach programu przewidziana została możliwość użycia kluczy. Pełnią one rolę zmiennych bulowskich i służą do sterowania programem. Działanie kluczy oraz wskazówki w której części programu mogą być użyte podane są w tablicy 3.

Tablica 3

Nr. klucza	działanie, wciśnięcie powoduje	Cz.I	Cz.II	komentarze
16	przygotowanie taśmy z układem współrzędnych biegunowych dla plottera		☒	umożliwia otrzymanie wykresów układu współrzędnych biegunowych
17	wczytanie z monitora maxmaxny	☒		umożliwia doliczanie dodatkowych fragmentów mapy synoptycz.
18	przygotowanie taśmy dla plottera		☒	
19	powoduje zmniejszenie skali wykresów		☒	umożliwia przejście od mapy dla 1/4 do mapy dla całej aperatury
20	zakończenie aktualnej pętli izowania		☒	umożliwia skrócenie czasu obliczeń
21	wykonanie obliczeń dla źródła izotropowego	☒	☒	obowiązuje jednakowy stan klucza w obu częściach programu
22	ominięcie wydruku wyników pośrednich	☒		umożliwia skrócenie czasu obliczeń
22	ominięcie wydruku wyników końcowych		☒	umożliwia skrócenie czasu obliczeń

Przy tłumaczeniu translatorem ta5 część pierwsza programu zajmuje 5098 , a część druga 3785 komórek w pamięci operacyjnej maszyny.

Program uruchamiamy przez wypisanie komunikatu w13 na monitorze.

Wydruki

W wyniku działania części I programu zostają wydrukowane na drukarce: tekst identyfikacyjny, dane wejściowe, dane wejściowe zmodyfikowane, wartości maksymalne modułów składowych E_x , E_y , E_z , argument $E_y(0,0)$ oraz tablice współrzędnych punktów siatki i występujące w tych punktach wartości składowych E_y i E_z pola oraz parametry elipsy polaryzacyjnej. Wartości składowej E_x drukowane są w oddzielnych tablicach tylko wtedy, gdy $\max |E_x|$ przybiera niepokojąco dużą wartość.

Część II programu powoduje ponowne wydrukowanie tekstu identyfikacyjnego i wartości maksymalnych, a następnie aktualnych wartości c_x , c_y , t_x , t_y występujących w procedurze DRAW, informacji o każdej kolejnej obliczanej izolacji i ciągu par współrzędnych jej punktów.

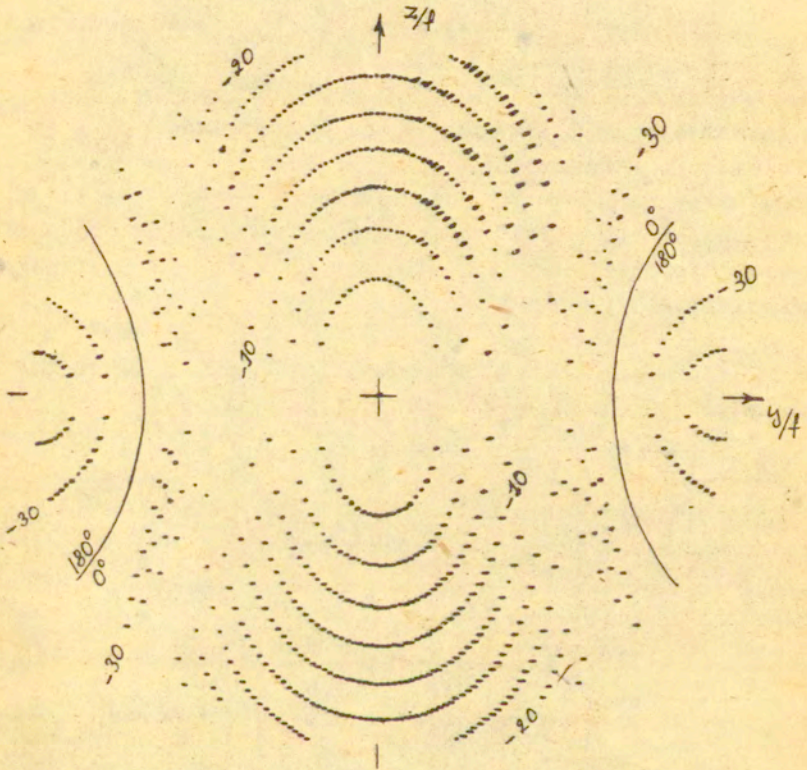
W czasie działania programu na monitorze drukowane są liczne informacje ułatwiające współpracę operatora z programem oraz umożliwiające opisywanie taśmy dla plottera.

Wyniki

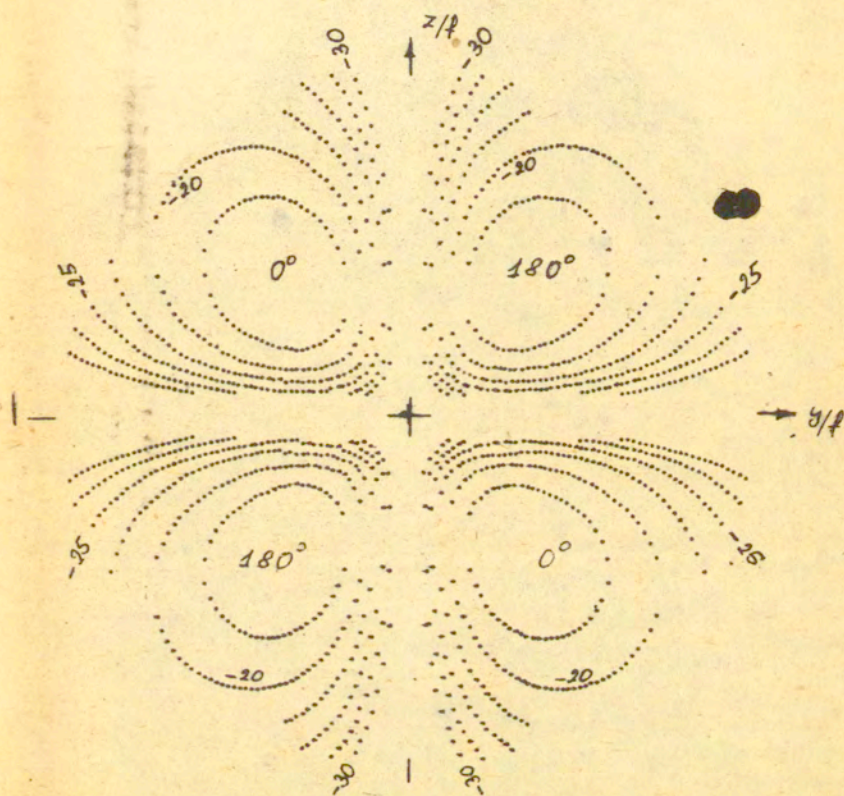
Poniżej przedstawione są rodziny „map synoptycznych” parametrów pola elektromagnetycznego w aperturze anteny obliczone za pomocą programu „APERTURE” dla podanych parametrów wejściowych.

Bibliografia

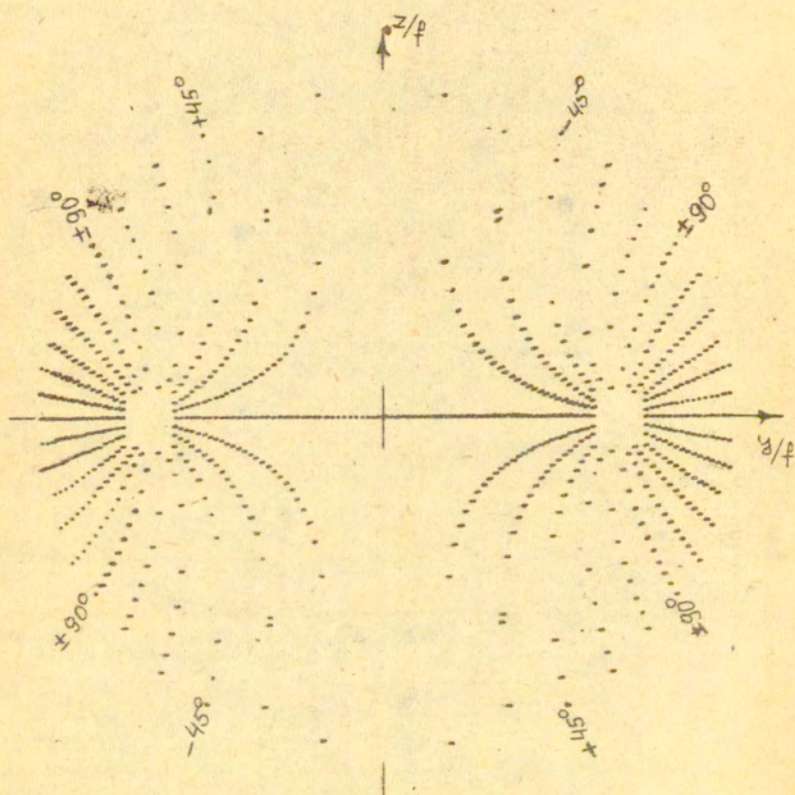
1. Rusch W.V.T., Potter P.D., Analysis of Reflector Antennas. Academic Press, New York - London 1970.
2. Bodnar D.G., Cross-Polarized Characteristics of Monopulse Difference Patterns, IEEE/AP-S Int.Symposium Quebec, vol.2, pp.477-480, 1980.
3. Trueman Ch.W., Pavlasek T.J.F., Kubina St.J., Parametric and Synoptic Charts for Determining the Patterns of an Elemental Dipole with a Scattering Mirror or Cylinder. IEEE Trans. EMC-19, No.4, pp.365-370, 1977.
4. Węgrowicz L.A., V-dipole Array as a Primary Feed for Deep Paraboloidal Reflectors. IEEE/AP-S International Symp. Quebec, vol.2, pp.706-708, 1980.



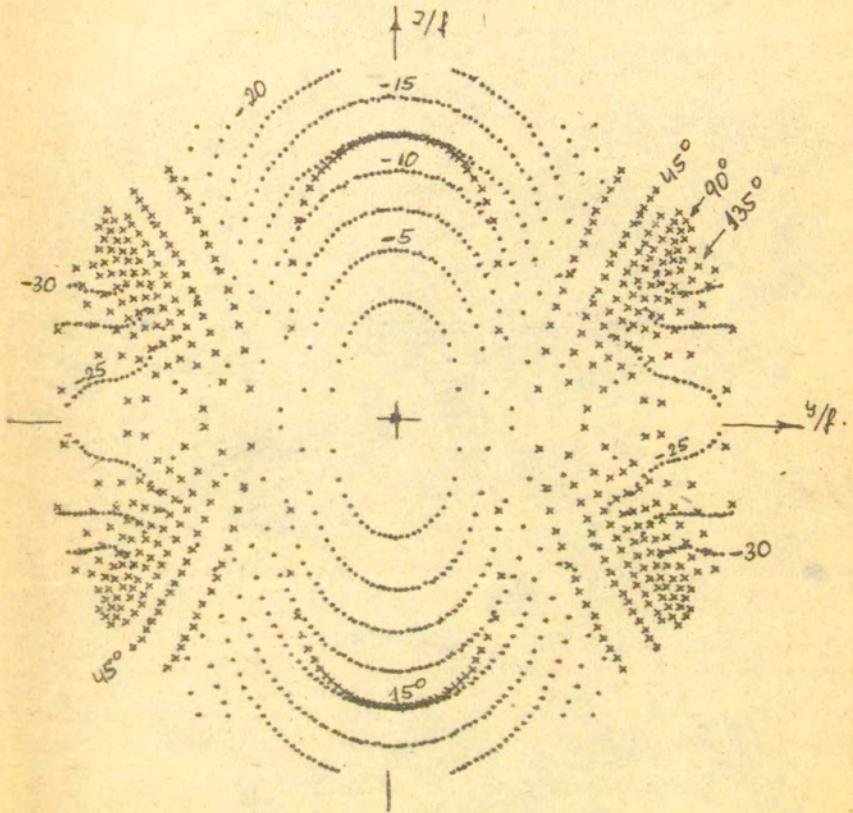
Rys.2.1. Mapa rozkładu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 180^\circ$ długość ramienia dipola $h = 0.25 \lambda$, odległość dipoli $l = 0.25 \lambda$, przesunięcie fazowe dipoli $\gamma = 90^\circ$. Mapa nie opracowana po wyjściu z plottera.



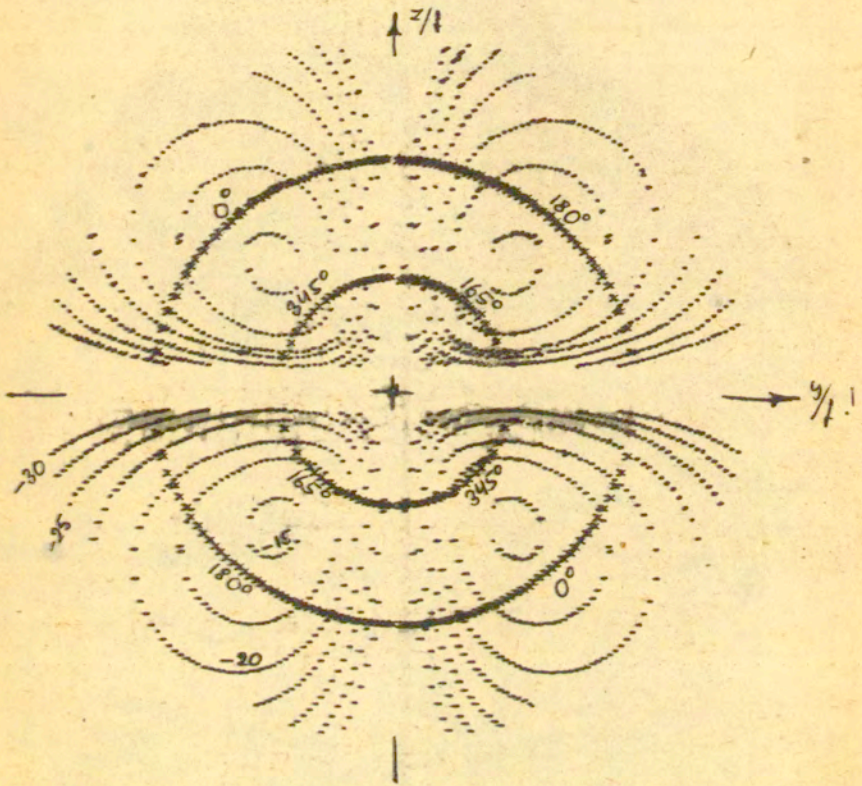
Rys.2.2. Mapa rozkładu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



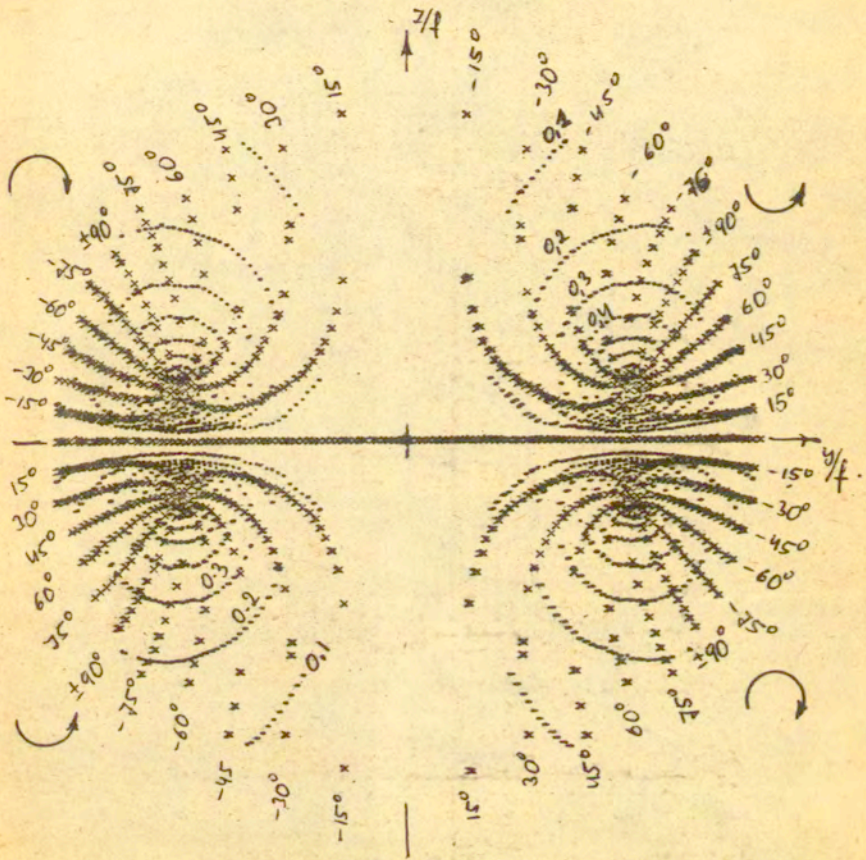
Rys.2.3. Mapa rozkładu kąta nachylenia α elipsy polaryzacyjnej pola na aperturze zwierciadła. Iloraz półosi elipsy $m = 0$.



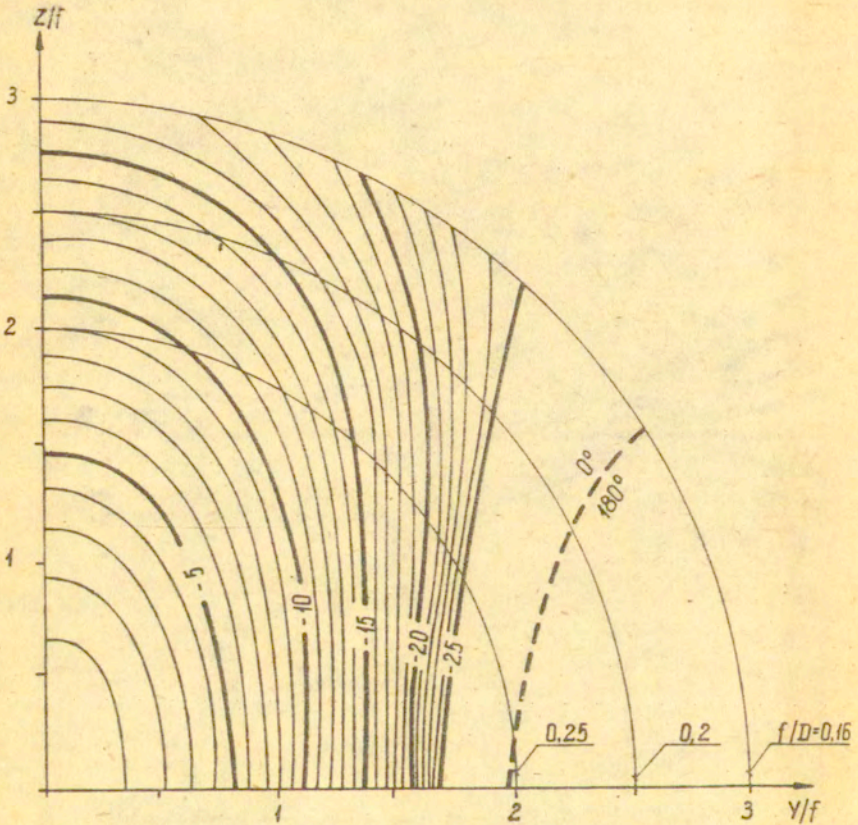
Rys.3.1. Mapa rozkładu modułu (•) i argumentu (x) składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\Psi = 135^\circ$, długość ramienia dipola $h = 0.25 \lambda$, przesunięcie fazowe dipoli $\gamma = 90^\circ$, odległość dipoli $l = 0.25 \lambda$.



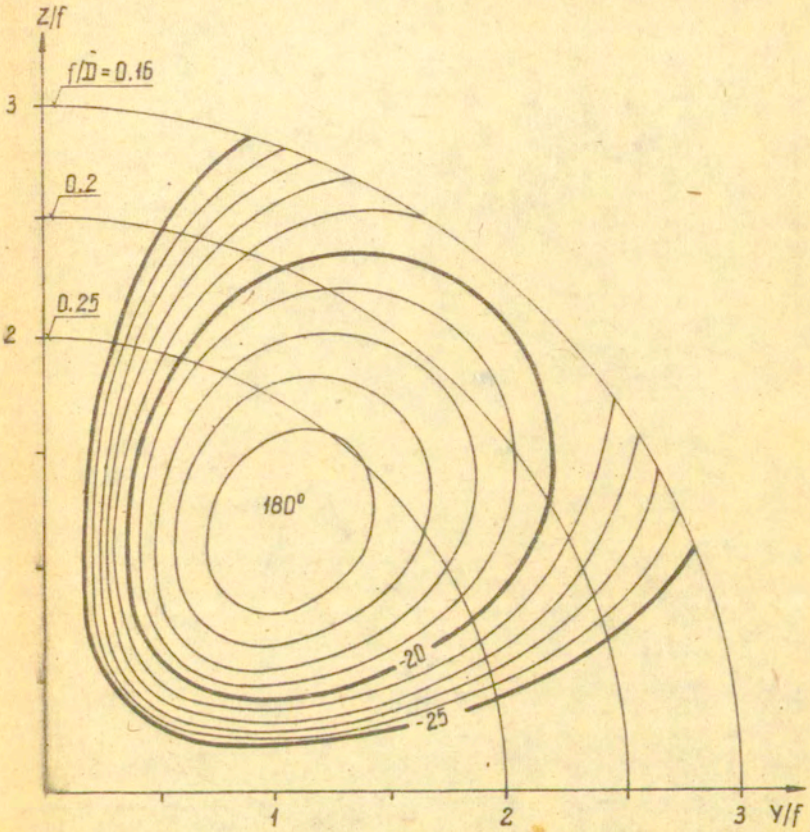
Rys.3.2. Mapa rozkładu modułu (·) i argumentu (x) składowej crosspolaryzacyjnej pola w aperturze zwierciadła.



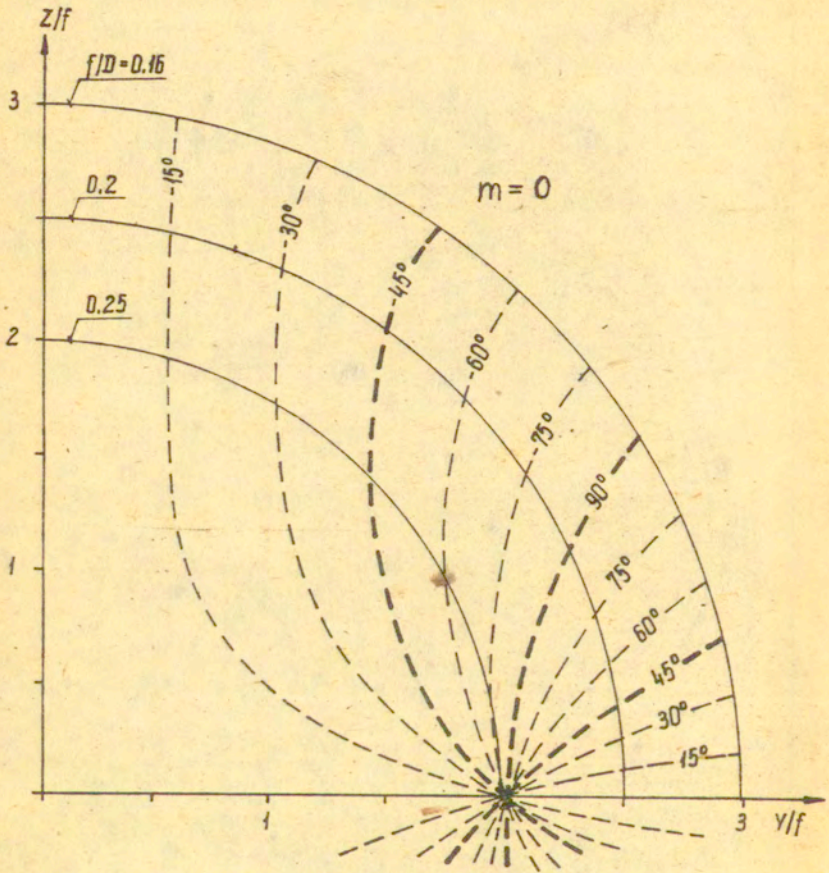
Rys.3.3. Mapa rozkładu współczynnika eliptyczności m (•) i kąta nachylenia (x) elipsy polaryzacyjnej pola na aperturze zwierciadła.



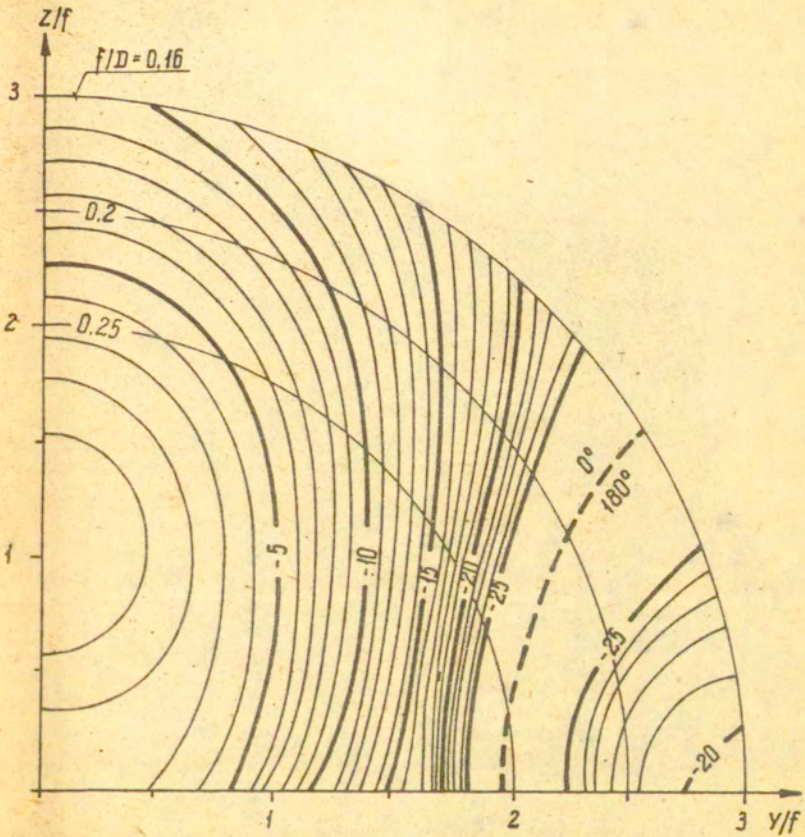
Rys.4.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 180^\circ$, dług. ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.25\lambda$, przesunięcie fazowe dipoli $\delta = 90^\circ$.



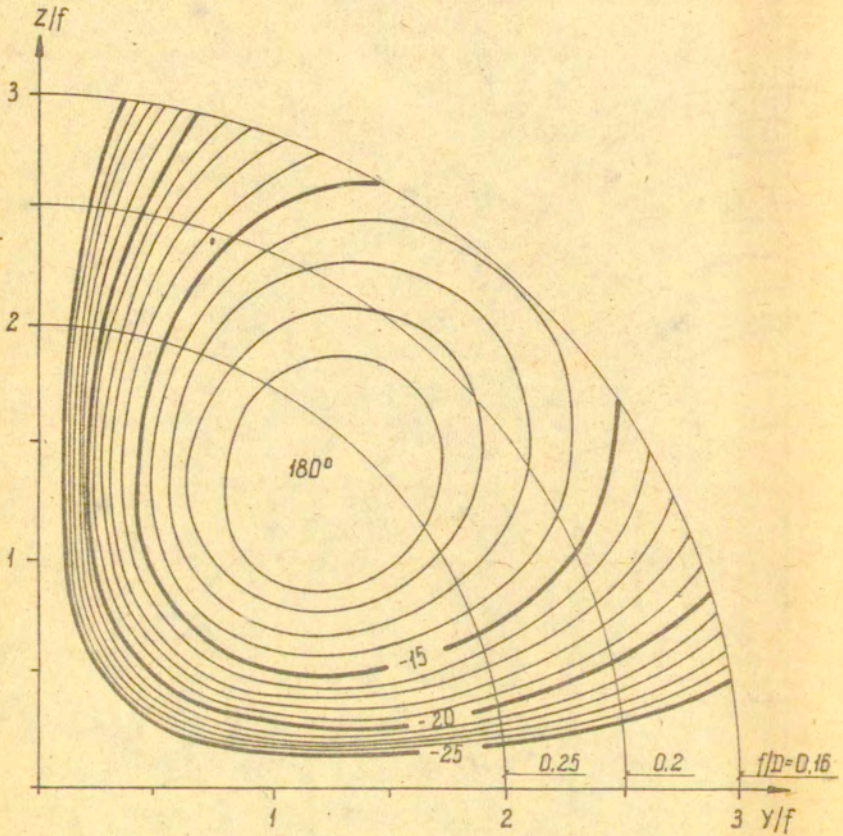
Rys.4.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



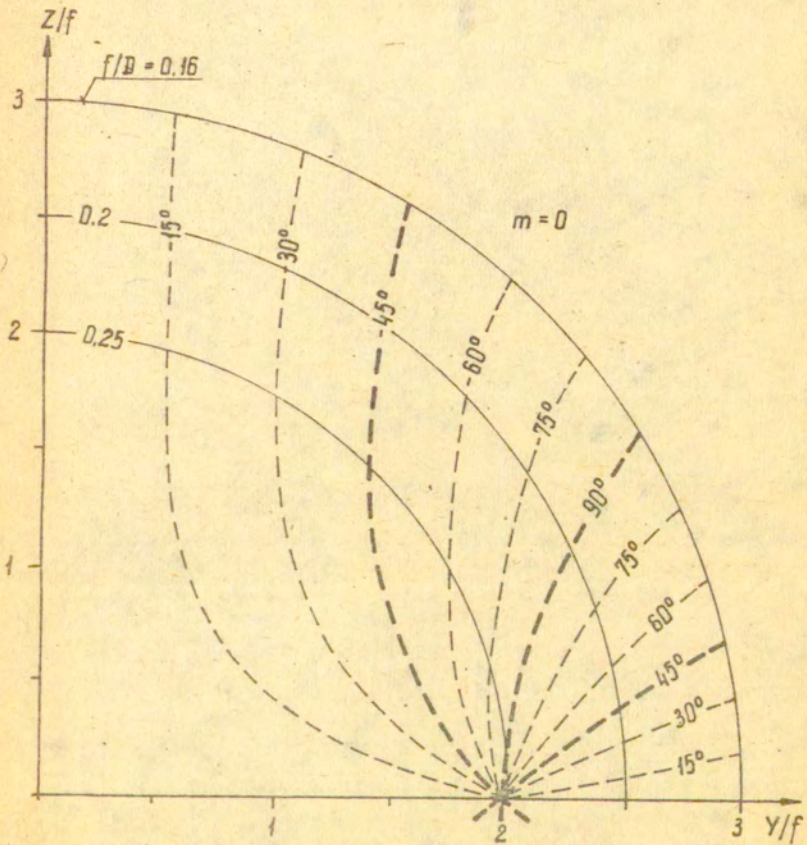
Rys.4.3. Mapa rozkładu wsp. eliptyczności m i kąta nachylenia α elipsy polaryzacyjnej pola na aperturze zwierciadła.



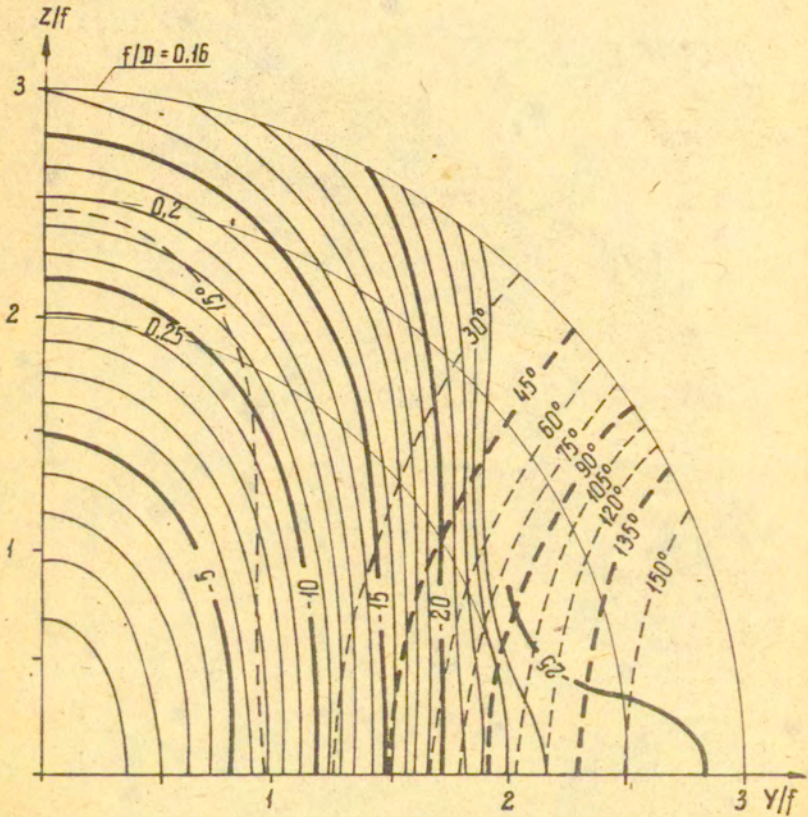
Rys.5.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 180^\circ$, długość ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.416\lambda$, przesunięcie fazowe dipoli $\gamma = 30^\circ$.



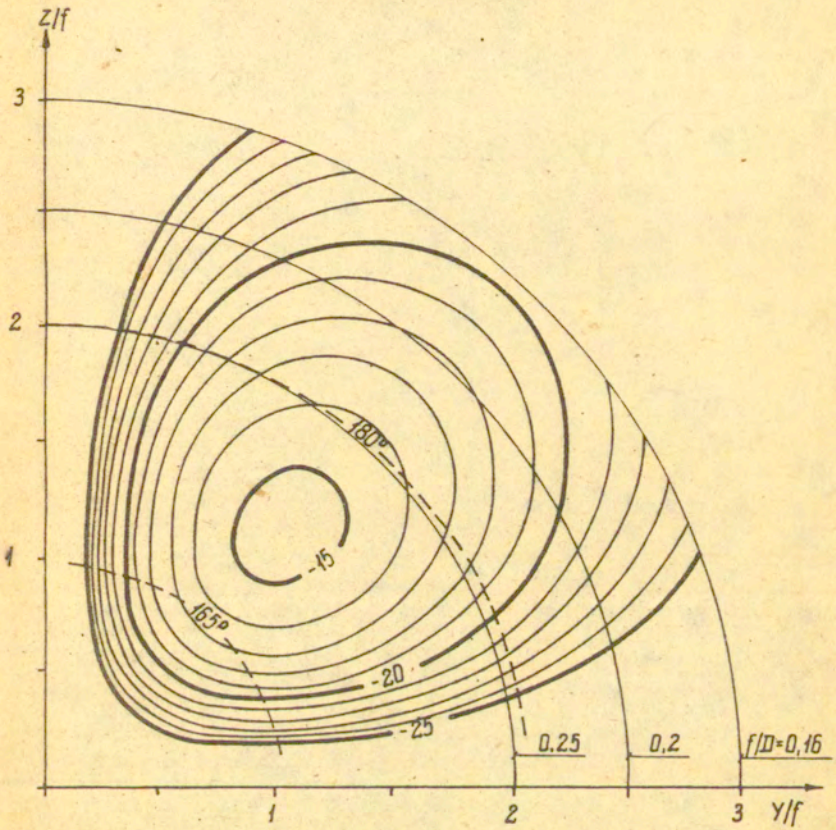
Rys.5.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



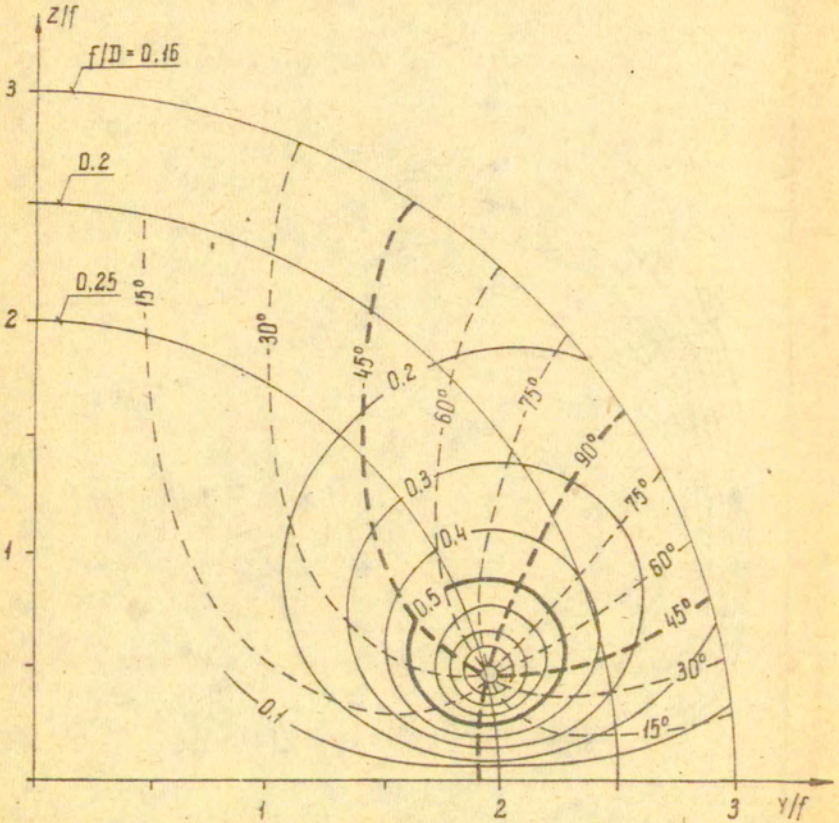
Rys.5.3. Mapa rozkładu wsp. eliptyczności m i kąta nachylenia α elipsy polaryzacyjnej pola na aperturze zwierciadła.



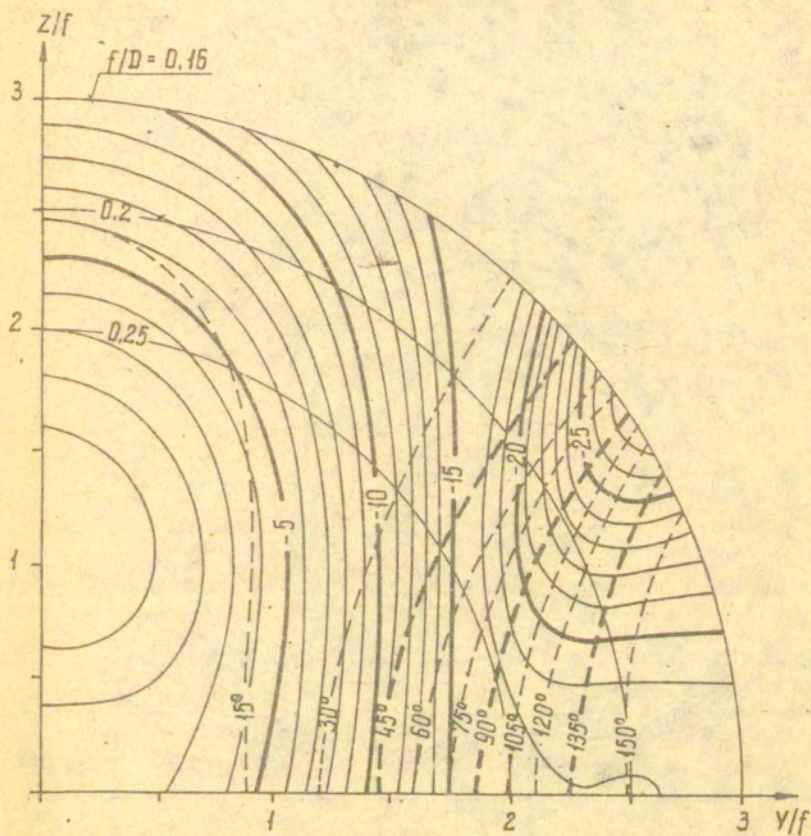
Rys.6.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 135^\circ$, dług. ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.25\lambda$, przesunięcie fazowe dipoli $\delta = 90^\circ$.



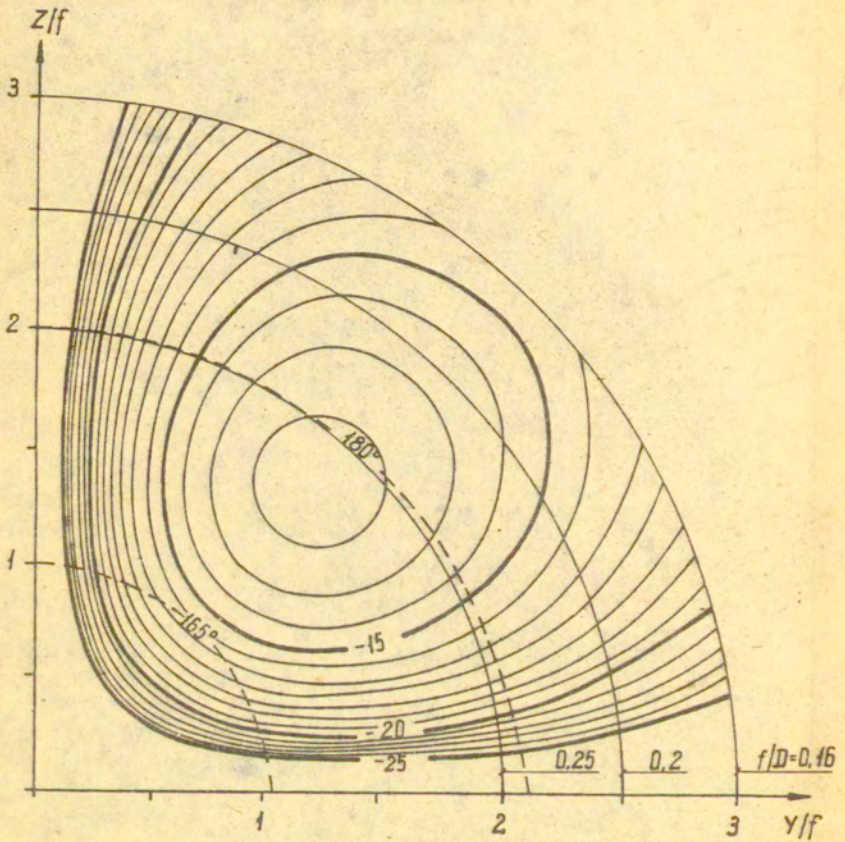
Rys.6.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



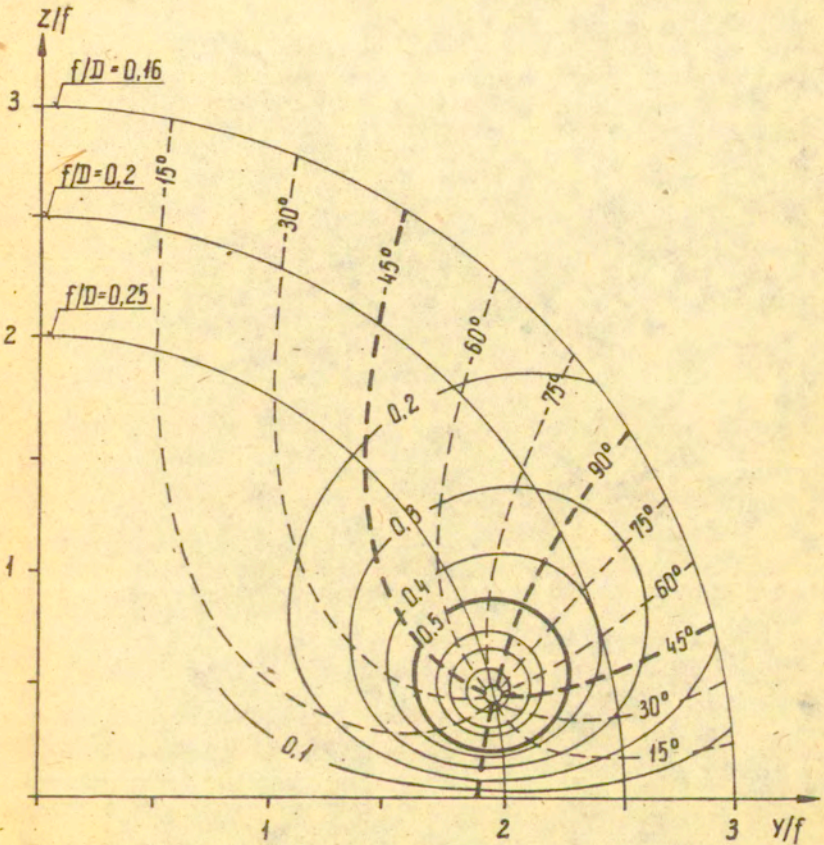
Rys.6.3. Mapa rozkładu wsp. eliptyczności m i kąta nachylenia α elipsy polaryzacyjnej pola na aperturze zwierciadła.



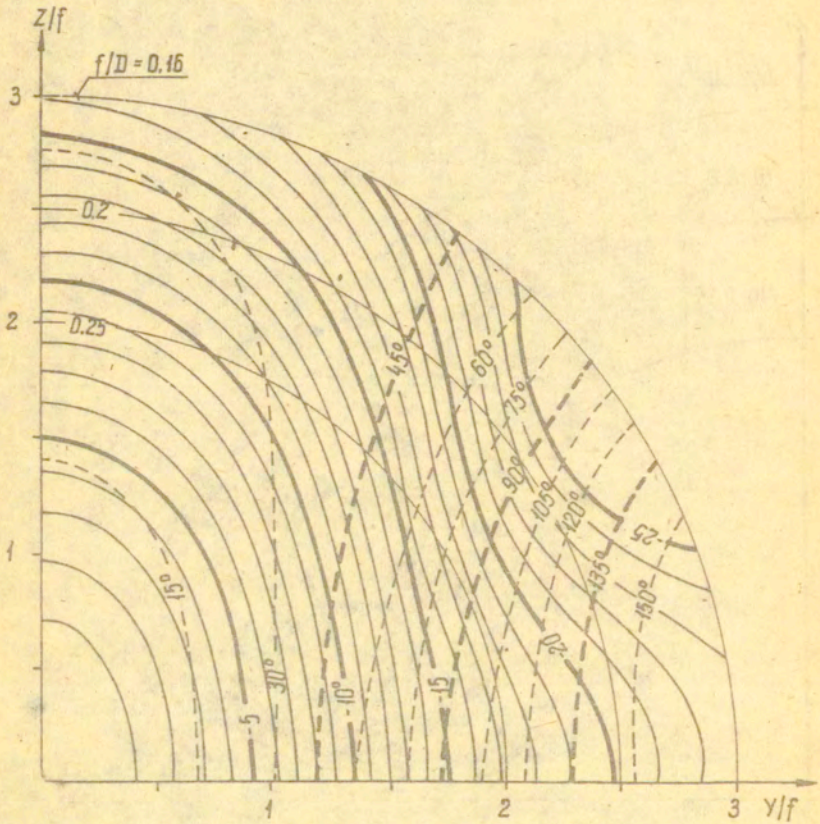
Rys.7.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 135^\circ$, dług. ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.416\lambda$, przesunięcie dipoli $\gamma = 30^\circ$.



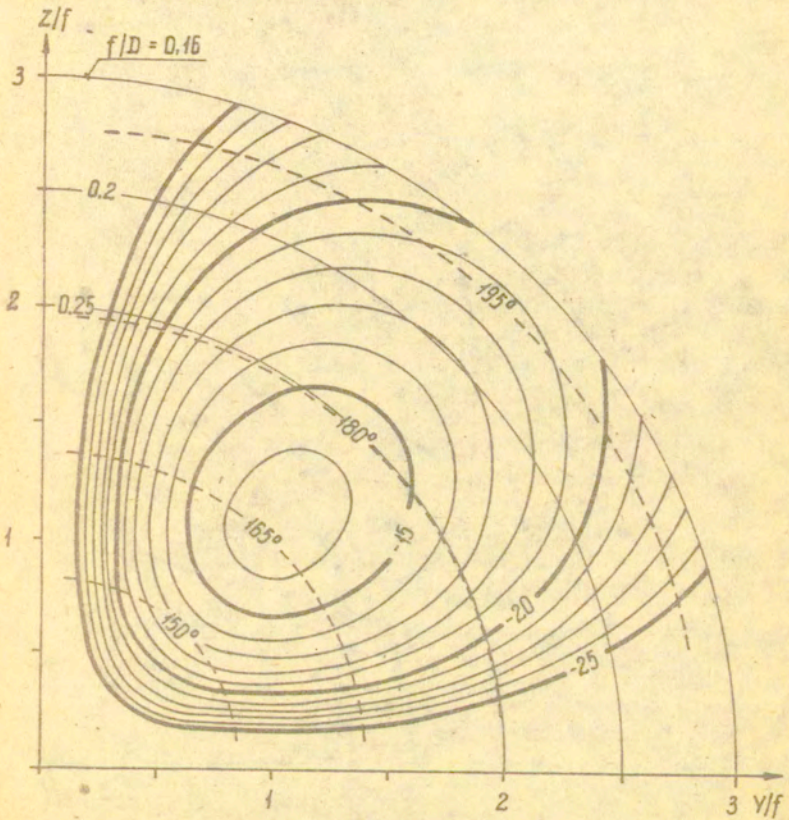
Rys.7.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



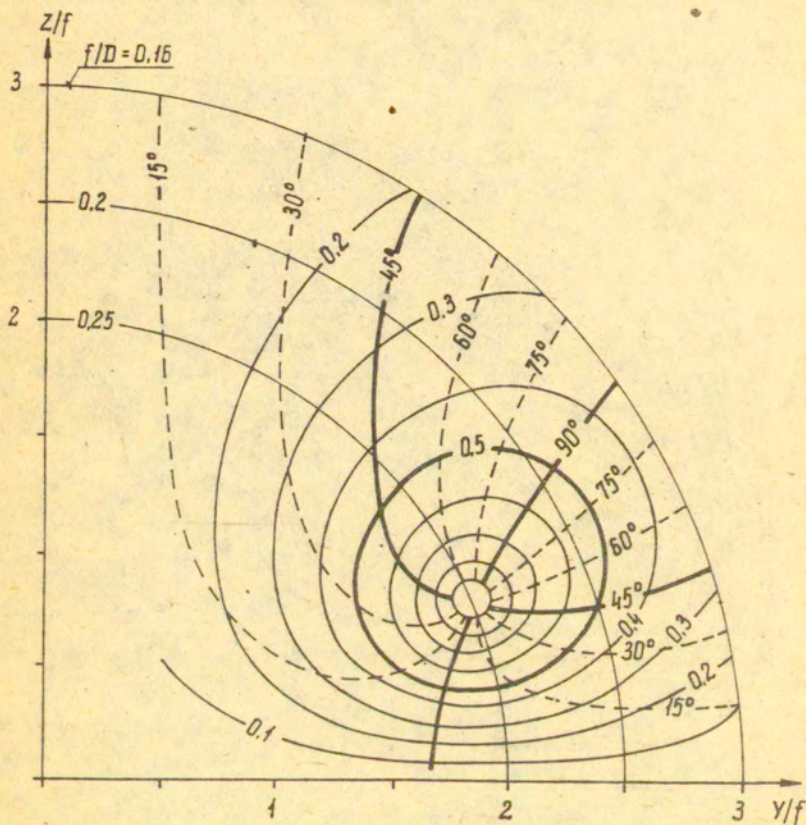
Rys. 7.3. Mapa rozkładu wsp. eliptyczności m i kąta nachylenia α elipsy polaryzacyjnej pola na aperturze zwierciadła.



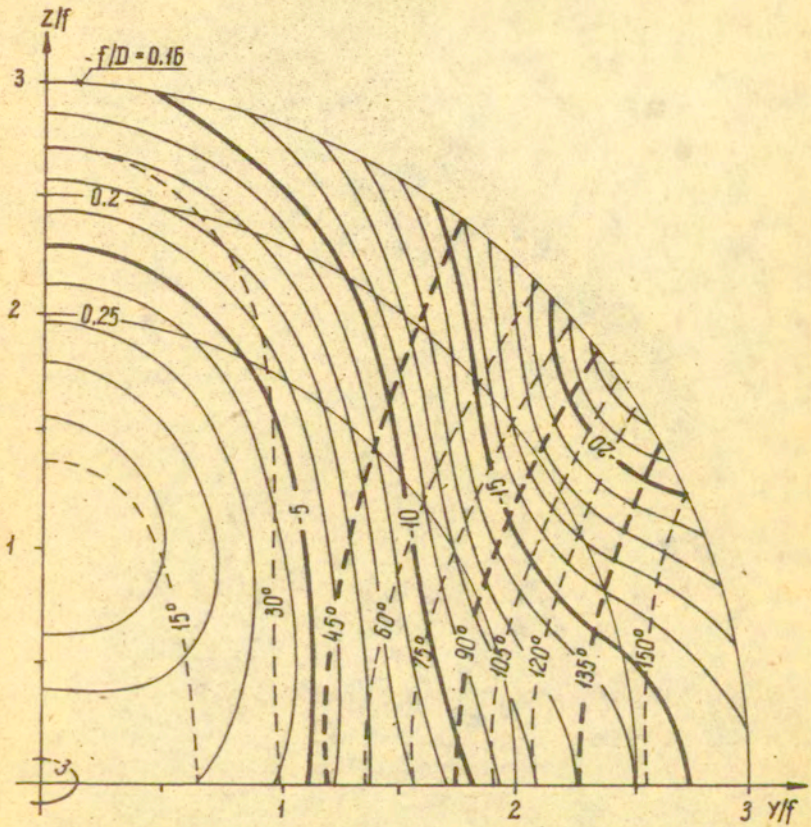
Rys.8.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 90^\circ$, długość ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.25\lambda$, przesunięcie fazowe dipoli $\chi = 90^\circ$.



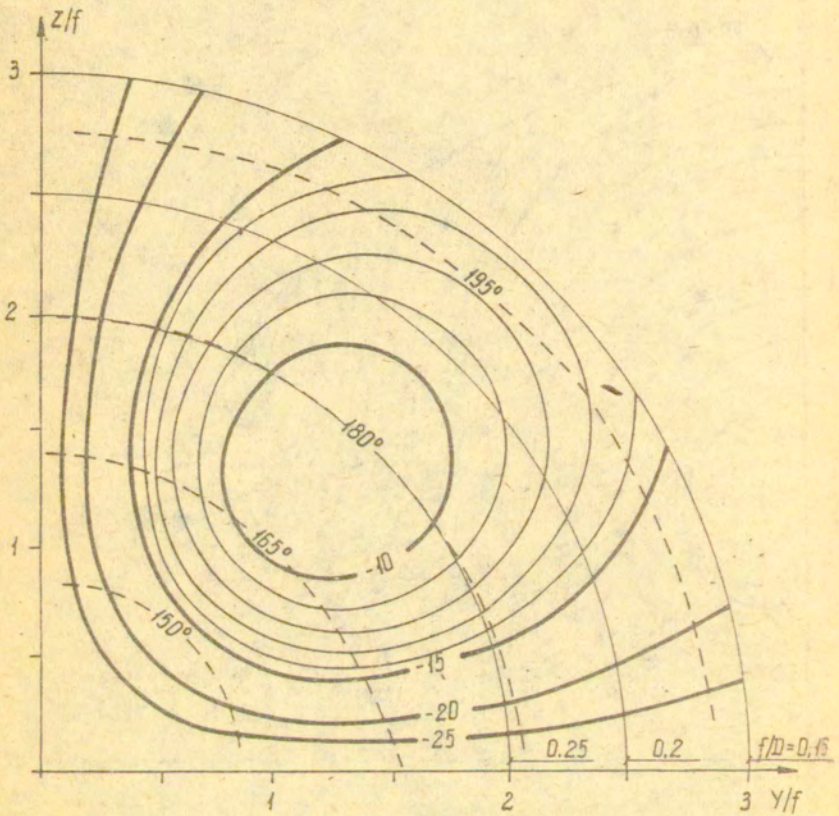
Rys. 8.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



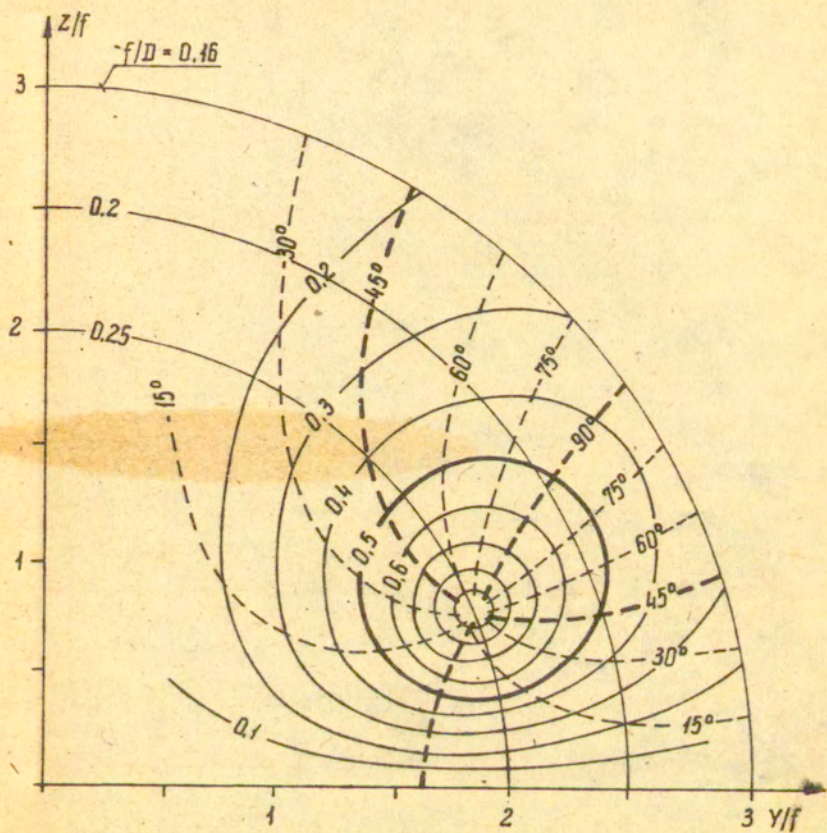
Rys.8.3. Mapa rozkładu wsp. eliptyczności m i kąta nachylenia χ elipsy polaryzacyjnej pola na aperturze zwierciadła.



Rys.9.1. Mapa rozkładu modułu i argumentu składowej podstawowej pola na aperturze zwierciadła. Kąt rozwarcia dipola $2\psi = 90^\circ$, długość ramienia dipola $h = 0.25\lambda$, odległość dipoli $l = 0.25\lambda$, przesunięcie fazowe dipoli $\gamma = 90^\circ$.



Rys.9.2. Mapa rozkładu modułu i argumentu składowej crosspolaryzacyjnej pola na aperturze zwierciadła.



Rys.9.3. Mapa rozkładu wsp. eliptyczności m i kąta θ nachylenia elipsy polaryzacyjnej pola na aperturze zwierciadła.

```

4  comment APERTURA<< z dn 19.03.1980 cz I;
begin
integer dpmx, dpmv, dpmz, dpax, dpay, dpaz, dpfi, dpdel, dpksi, il, ilp,
dpth, dpm, dnal, dpst, i, j, iy, jz, ster, k, k, dpv, dpiy;
real pi, grd, dls, dlug, ps, ps2, yp, dy, yk, y, zp, dz, zk, z, THRzer,
Q, R, ksi, theta, fi, delta, mr, mrr, mp, mpp, THR, THP,
mod, faz, a1, a2, mx, my, mz, argx, argy, argz, maxmaxxy,
m, alfa, c4, c, lnumx, lnumy, lnumz, odl, op, fz, prs,
RR, maxmaxmx, maxmaxmz, ro, cni, lnum,
izop, krizo, izok, izo, krap, krap1, krap2,
krap3, krap4, krap5;
integer array TEXT[1:30], JZTY[1:2];
array V[1:10];
44 procedure COMPLEX(res, ims, mod, arg);
value res, ims;
real res, ims, mod, arg;
comment oblicza modul oraz argument w radianach (0, 2*pi)
liczby zespolonej res + i*ims;
begin
mod:=sqrt(res*res+ims*ims);
if mod gt .v-60 then
begin
arg:=abs(ims/mod);
if arg gt 1.0 then arg:=1.0;
arg:=arcsin(arg);
if res lt .0 then arg:=pi-sign(ims)*arg;
if ims lt .0 and res ge .0 then arg:=2.0*pi-arg;
end else mod:=arg:=.0;
if abs(arg-2.0*pi) lt v-6 then arg:=.0;
end COMPLEX;
34 procedure TRANS(y, z, ksi, theta);
value y, z; real y, z, ksi, theta;
comment oblicza wspolrzedne sferyczne ksi (0, 2*pi) i theta (0, pi)
kierunku, z ktorego widoczny jest z ogniska punkt

```

na aperturze o wspolrzędnych prostokatnych y, z unormowanych do F. Biegun (0,0) w kierunku dodatnich x;

```

begin
  real md,u;
  md:=y*x+z*x;
  if md <= -30 then
  begin
    theta:=2.0*arctan(sqrt(md)/2);
    u:=sqrt(y*y/md);
    if u <= 1.0 then u:=1.0;
    ksi:=arcsin(u);
    if z <= 0 then ksi:=pi-sign(y)*ksi;
    if y <= 0 and z <= 0 then ksi:=2.0*pi-ksi;
  end
  end TRANS1;
50 procedure TRANS3(ksi, theta, fi, delta);
  value ksi, theta; real ksi, theta, fi, delta;
  comment oblicza wspolrzędne geograficzne fi(-pi,pi) i delta(-pi/2, pi/2)
  punktu o wspolrzędnych sferycznych ksi(0,2*pi) i theta(0,pi);
  begin
    real pip, sks, cks, sth, cth, rob;
    pip:=arcsin(1.0);
    sks:=sin(ksi); cks:=cos(ksi); sth:=sin(theta); cth:=cos(theta);
    if abs(theta-pip) <= .00001745 then
    begin
      rob:=sth*sks/sqrt(cth*cth+sth*sks*sth*sks);
      if rob <= 1.0 then rob:=1.0;
      fi:=arcsin(rob);
      delta:=arcsin(sth*sks);
      if theta <= pip then
      fi:=if ksi <= 2*pip then 2*pip-fi else -2*pip-fi;
    end
  end
  begin
    fi:=if abs(sks) <= .00001745 then .0 else sign(sks)*pip;
    delta:=arcsin(cks);
  end
  end TRANS3;

```

```
72 procedure TRANS4(chi,ro,ksi,theta);
   value chi,ro;
   real chi,ro,ksi,theta;
   comment Oblicza wspolrzedne sferyczne ksi(0,2*pi),
           theta(0,pi) kierunku pod ktorym jest widac
           z ogniska punkto wspolrzednych chi(0,2*pi),
           ro(0, niesk) na aperturze;
   begin
   real r;
   r:=ro*ro; ksi:=chi;
   theta:=arccos((4.0-r)/(4.0+r));
   end TRANS4;
84 real procedure FAZA ZERO;
   comment faza skladowej rownoleglej dla ksi=0, theta=0;
   begin
   real acp; acp:=dlg*cos(ps);
   COMPLEX(2.0*cos(acp)-2.0*cos(dlg),-2.0*sin(acp)+2.0*cos(ps)*sin(dlg),cp,fz);
   FAZA ZERO:=-fz;
   end FAZA ZERO;
91 procedure MNOZ SIAT(theta,mod,faz);
   value theta; real theta,mod,faz;
   comment mnoznik siatki dwuch elementow,
           uwaga: faza zero ma byc rowna .0;
   if key(21) then begin mod:=1.0; faz:=.0 end else
   COMPLEX(2.0*cos(pi*odi*(cos(theta)-prs)).0,mod,faz);
97 procedure MN REFL(mr,mp,ksi,theta,fi,delta,THR,THP,
                    mx,my,mz,argx,argy,argz);
   value mr,mp,ksi,theta,fi,delta,THR,THP;
   real mr,mp,ksi,theta,fi,delta,THR,THP;
   mx,my,mz,argx,argy,argz;
   comment Realizuje Eoddb=(n.Epad).n -(n.Epad)*n, gdzie
           mr, THR - skladowa rownoleznikowa pola padajacego,
           mp, THP - skladowa poludnikowa pola padajacego;
   begin
   real sf,cf,sd,cd,st,ct,sk,ck,sdsf,sdcf,att,stsk,ctck,
           skck,ctck,ctsk,ct1,ctr,ctf,ctp,rnr,imr,rmp,imp,
```

```

rExp, iExp, rEyp, iEyp, rEzp, iEzp, rExo, iExo, rEyo, iEyo, rEzo, iEzo;
sf:=sin(fi); cf:=cos(fi); sd:=sin(delta); cd:=cos(delta);
st:=sin(theta); ct:=cos(theta); sk:=sin(ksi); ck:=cos(ksi);
sdfs:=sd*sf; sdfc:=sd*cf;
if theta le krap then att:=(1.0+ct)/2 else
begin mx:=my;mz:=argx:=argy:=argz:=.0; goto fin end;
stsk:=st*sk; stck:=st*ck; skck:=sk*ck; ctck:=ct*ck;
ctsk:=ct*sk; ct1:=1.0-ct;
str:=sin(THR); ctr:=cos(THR); stp:=sin(THP); ctp:=cos(THP);
rnr:=mr*ctr*att; imr:=mr*str*att;
rmp:=mp*ctp*att; imp:=mp*stp*att;
rExp:=-rnr*sf+rmp*sdcf;
iExp:=-imr*sf+imp*sdcf;
rEyp:= rnr*cf+rmp*sdfs;
iEyp:= imr*cf+imp*sdfs;
rEzp:=-rmp*cd;
iEzp:=-imp*cd;
rExo:=rExp*ct+rEyp*stsk+rEzp*stck;
iExo:=iExp*ct+iEyp*stsk+iEzp*stck;
rEyo:=rExp*stsk+rEyp*(ctsk*sk+ck*ck)+rEzp*ct1*skck;
iEyo:=iExp*stsk-iEyp*(ctsk*sk+ck*ck)+iEzp*ct1*skck;
rEzo:=rExp*stck+rEyp*ct1*skck-rEzp*(ctck*ck+sk*sk);
iEzo:=iExp*stck+iEyp*ct1*skck-iEzp*(ctck*ck+sk*sk);
COMPLEX(rExo, iExo, mx, argx);
COMPLEX(rEyo, iEyo, my, argy);
COMPLEX(rEzo, iEzo, mz, argz);
fin;
end MN REFL;
436 real procedure PRZEDZIAL(x);
value x; real x;
comment sprowadza katy do przedzialu (0, 2*pi);
begin
real pi2; integer n;
pi2:=4.0*arcsin(1.0); n:=entier(abs(x)/pi2);
PRZEDZIAL:=if x lt .0 then x+(n+1)*pi2 else x-n*pi2
end PRZEDZIAL;

```

```
44 procedure PRIMESOURCE(fi, delta, mrrp, mpp, THR, THP);
   value fi, delta; real fi, delta, mrrp, mpp, THR, THP;
   comment Oblicza wartosci skladowych zespolonych
   rownoznacznikowych mrrp i poludnikowych mpp
   anteny V;
begin
   real sps, fmp, fpp, sfmp, cfmp, sfpp, cfpp, sd, cd, sdlg, cdlg,
   mian, re1, re2, im1, im2, rer, imr, rep, imp;
42 procedure POMPRIME(c, re, im);
   value c; real c, re, im;
begin
   re:=(cos(dlg<cd>c)-cdlg)/mian;
   im:=(sin(dlg<cd>c)-cd<cdlg)/mian
end POMPRIME;
   if key(21) then
begin
   mrrp:=1.0; mpp:=THP:=.0;
   THR:=if abs(fi) le pi/2.0 then .0 else pi
end else
begin
   sps:=sin(ps);
   fmp:=fi-ps; fpp:=fi+ps;
   sfmp:=sin(fmp); cfmp:=cos(fmp);
   sfpp:=sin(fpp); cfpp:=cos(fpp);
   sd:=sin(delta); cd:=cos(delta);
   sdlg:=sin(dlg); cdlg:=cos(dlg);
   mian:=1.0-cd<cfmp>cd<cfmp;
   if mian lt e4 then re1:=im1:=.0 else
   POMPRIME(cfmp, re1, im1);
   mian:=1.0-cd<cfpp>cd<cfpp;
   if mian lt e4 then re2:=im2:=.0 else
   POMPRIME(cfpp, re2, im2);
   rer:=sps<(-sfmp<re1+sfpp<re2);
   imr:=sps<(-sfmp<im1-sfpp<im2);
   COMPLEX(rer, imr, mrrp, THR); THR:=PRZEDZIAL(THR-THRzer);
   rep:=sps<sd<(cfmp<re1-cfpp<re2);
```

```
imp:=-sps*sd*(cfmp*im1-cfpp*im2);
COMPLEX(rep,imp,mpp,THP); THP:=PRZEDZIAL(THP-THRZER);
end key 21;
end PRIMESOURCE;

484 procedure POLELIPS1(Eg, Ed, THg, THd, m, alfa, ster);
value Eg, Ed, THg, THd;
real Eg, Ed, THg, THd, m, alfa; integer ster;
comment Oblicza parametry elipsy polaryzacyjnej,
      Eg - pole glowne, Ed - pole dodatkowe,
      alfa(-90,90) - kat miedzy Eg a najblizsza duza polosia elipsy,
      m(0,1) - stosunek osi
      alfa=360 oznacza nieokreslona faze;
begin
  e,grd,theta,sint, cost,g;
  real e,grd,theta,sint, cost,g;
  ei:=w-5; grd:=arctan(1.0)/45.0;
  theta:=THg-THd; sint:=sin(theta); cost:=cos(theta);
  if Eg lt e and Ed lt e then
    begin
      m:=0.0; alfa:=360.0*grd; ster:=6; goto kon;
    end;
  if Ed lt e then
    begin
      m:=100.0; alfa:=0; ster:=3; goto fin;
    end;
  if Eg lt e then
    begin
      m:=100.0; alfa:=sign(cost)>90.0*grd; ster:=3; goto fin;
    end;
  if abs(Eg-Ed) lt e then
    begin
      if abs(cost) lt e then
        begin
          m:=1.0; alfa:=360.0*grd;
          ster:=if sint gt .0 then 5 else 1; goto fin;
        end;
      if abs(sint) lt e then
```

```
begin
mi:=100.0; alfa:=sign(cost)*45.0*grd; sters:=3; goto fin;
end;
g:=(Eg*Eg/(Ed*Ed)+2.0*cost*cost+Ed*Ed)/(Eg*Eg)/((sint*sint);
mi:=sqrt(.5*(-g+sqrt(g*g-4.0)));
alfa:=sign(cost)*45.0*grd;
sters:=if sint gt .0 then 4 else 2;
goto fin;
end;
if abs(sint) lt .2 then
begin
mi:=100.0; sters:=3;
alfa:=.5*arctan(2.0*Eg*Ed*cost/(Eg*Eg-Ed*Ed));
if Ed gt Eg then alfa:=-sign(alfa)*90.0*grd+alfa;
goto fin;
end;
g:=-((Eg*Eg/(Ed*Ed)+2.0*cost*cost+Ed*Ed)/(Eg*Eg)/((sint*sint);
mi:=sqrt(.5*(-g+sqrt(g*g-4.0)));
alfa:=.5*arctan(2.0*Eg*Ed*cost/(Eg*Eg-Ed*Ed));
if Ed gt Eg then alfa:=-sign(afa)*90.0*grd+afa;
sters:=if sint gt .0 then 4 else 2;
fin;
m:=if m gt 99.9 then .0 else 1.0/m;
if m lt .005 then sters:=3;
kon:
end POLLEIPS1;
242 procedure TODR(dp,n,A);
integer dp,n; array A;
comment Przesyła na bebena od adresu dp
n elementowa real array A;
begin
drumplace:=dp; to drum(n,A[1]); dp:=drumplace
end TODR;
249 procedure FROMDR(dp,n,A);
integer dp,n; array A;
comment Sciaga z bebna od adresu dp
```



```
n elementowa real array A;
begin
drumplace:=dp; from drum(n,A[1]); dp:=drumplace
end FROMDR;
256 procedure ITODR(dp,n,A);
integer dp,n; integer array A;
comment Przesyla na beben od adresu dp
n elementowa integer array A;
begin
drumplace:=dp; to drum(n,A[1]); dp:=drumplace
end ITODR;
263 procedure IFROMDR(dp,n,A);
integer dp,n; integer array A;
comment Sciaga z bebna od adresu dp
n elementowa integer array A;
begin
drumplace:=dp; from drum(n,A[1]); dp:=drumplace
end IFROMDR;
270 procedure DP;
comment Ustala adresy na bebnie;
begin
dpmx:=0; dpmv:=10000; dpmz:=20000;
dpax:=30000; dpay:=40000; dpaz:=50000;
dpf1:=60000; dpks1:=70000; dpdel:=80000;
dpth:=90000; dpm:=100000; dpal:=110000;
dpst:=120000; dpv:=130000; dply:=130050;
end DP;
279 procedure STERO(i);
value i; integer i;
comment Wspolpracuje z procedura POLELIPS1;
if i=1 then print('kle') else
if i=2 then print('ele') else
if i=3 then print('lin') else
if i=4 then print('epr') else
if i=5 then print('kpr') else print('zer');
287 procedure AUXCARD(zp,dz,zk,jz,yp,dy,yk,ly);
```

```

integer jz, iy; real zp, dz, zk, yp, dy, yk;
comment. Modyfikuje dane dla procedury CARD;
begin
  integer l;
  if zp le .0 then
  begin
    l:=entier(-zp/dz+.5);
    zp:=if 2*(l div 2)=1 then -(l+1)*dz else -l*dz;
  end;
  if yp le .0 then
  begin
    l:=entier(-yp/dy+.5);
    yp:=if 2*(l div 2)=1 then -(l+1)*dy else -l*dy;
  end;
  jz:=entier((zk-zp)/dz+.5)+1;
  if 2*(jz div 2)=jz then jz:=jz+1;
  jz:=jz-1;
  zk:=zp+jz*dz;
  iy:=entier((yk-yp)/dy+.5)+1;
  if 2*(iy div 2)=iy then iy:=iy+1;
  iy:=iy-1;
  yk:=yp+iy*dy;
  iy:=iy+1; jz:=jz+1;
end AUXCARD;
32) pi:=2.0*arcsin(1.0); grd:=pi/180.0;
   c1:=8.685889638; c4:=v-5;
   setinput(0); setoutput(0); line(2);
   print('tekst'); instr(TEXT[1]); line(1);
   print('kat rozw dipola (stopnie)='); read(ps2); line(1);
   print(' .5 kata rozw apert (stopnie)='); read(krap); line(1);
   print(' dlugosc ramienia (lambda)='); read(dlug); line(1);
   print(' odlegosc dipoli (lambda)='); read(odl); line(1);
   print(' przes. fazowe dipoli (alfa/kc)='); read(prs); line(1);
32) SIATKA;
   print('y od '); read(yp); print(' oo '); read(dy);
   print(' do '); read(yk); line(1);

```

```

print('z od '); read(zp); print(' co '); read(dz);
print(' do '); read(zk); line(1);
wait('k17-maxmaxmy wczytane, k21-izotr, k22-nie tabl?');
ps1=.5*ps2*grd; dlG=2.0*pi*dlug; R1=2.0*tan(krap*grd/2.0);
RR1=R*R;
setoutput(3); line(4);
print('PROGRAM APERTURA>> Cz I??');
outstring(TEXT[1]); line(3);
print('kat rozwarcia dipola ps2 =');
format('_123.1...stopnie?'); print(ps2);
format('kat rozwarcia apertury krap==123.1...stopnie?');
print(2*krap);
format('promien apertury.....r/f...=123.123?'); print(R);
format('dlugosc ramienia.....dlug...=123.123-lambda?');
print(dlug);
format('odleglosc dipoli.....odl...=123.123-lambda?');
print(odl);
format('przesuniecie fazowe.....prsu...=123.123?');
print(prs); line(2);
DP;
AUXCARD(zp, dz, zk, jz, yp, dy, yk, iy);
if iy*jz EQ 5000 then
begin setoutput(0); print('za gesta siatka?'); KOJO SIATKA end;
TRANS4(.0, zp, ksi, krap1);
TRANS4(.0, zk, ksi, krap2);
TRANS4(.0, yp, ksi, krap3);
TRANS4(.0, yk, ksi, krap4);
krap5=if krap1 EQ krap2 then krap1 else krap2;
krap5=if krap5 EQ krap3 then krap5 else krap3;
krap5=if krap5 EQ krap4 then krap5 else krap4;
krap:=krap*grd;
krap=if krap EQ krap5 then krap else krap5;
format('krap-zmodyfikowane.....=123.1...stopnie?');
print(2*krap*grd);
format('promien-zmodyfikowany.....=123.123?');
print(2.0*tan(krap/2.0));

```

343

prawnikowe [arowe

$$\delta = 360 * pns * ode ; [stopnie] \times$$

360

```

dlg:=2.0<pi>dlug; psi:=ps2<grad/2;
THRzer:=FAZA ZERO;
format('THR(0,0).....=-123.123-stopnie?');
print(PREDZIAL(THRzer)/grad); line(1);
V[1]:=vp; V[2]:=dy; V[3]:=yk; V[10]:=2.0<tan(krap/2.0);
V[4]:=zp; V[5]:=dz; V[6]:=zk; JZIV[1]:=jz; JZIV[2]:=fy;
krap:=krap+5.0<grad;
begin
369 array MX,MY,MZ,AX,AZ,AY,TH,DELTA,
      KSI,TH,M,ALFA[1:jz];
integer array STER[1:jz];
if key(21) then print('zrodlo izotropowe??');
y:=yp-dy; maxxmx:=maxmaxmy:=maxmaxmz:=100.0;
for i:=1 step 1 until iy do
begin
y:=y+dy; z:=zp-dz;
for j:=1 step 1 until jz do
begin
z:=z+dz;
TRANS1(y,z,ksi,theta); KSI[j]:=ksi; TH[j]:=theta;
TRANS3(ksi,theta,fi,delta); FI[j]:=fi; DELTA[j]:=delta;
PRIMESOURCE(fi,delta,mr,mp,THR,THP);
MNOZ SIAT(theta,mod,faz);
mrp:=mr<mod; mpp:=mp<mod; a1:=THR+faz; a2:=THP+faz;
MN REFL(mrp,mpp,ksi,theta,fi,delta,a1,a2,
      mx,my,mz,arx,arq,arz);
MX[j]:=mx; MY[j]:=my; MZ[j]:=mz;
AX[j]:=PREDZIAL(argx-pi);
AY[j]:=PREDZIAL(argy-pi);
AZ[j]:=PREDZIAL(argz-pi);
maxmaxmx:=if mx gt maxmaxmx then mx else maxmaxmx;
maxmaxmy:=if my gt maxmaxmy then my else maxmaxmy;
maxmaxmz:=if mz gt maxmaxmz then mz else maxmaxmz;
399 POLELIPS(my,mz,arq,arz,m,alfa,ster);
M[j]:=m; ALFA[j]:=alfa; STER[j]:=ster;
end;
TODR(dpmx,jz,MX); TODR(dpmz,jz,MZ);

```

```

402 TODR(dpax,jz,AX); TODR(dpay,jz,AY); TODR(dpez,jz,AZ);
TODR(dpf1,jz,FI); TODR(dpd1,jz,DELTA); TODR(dpth,jz,TH);
TODR(dpkst,jz,KSI); TODR(dpm,jz,M);
TODR(dpal,jz,ALFA); ITODR(dpst,jz,STER);
end i;
V[7]=maxmaxmx; V[9]=maxmaxmz;
if key(17) then
begin
setoutput(0); setinput(0); print('maxmaxmy=');
read(maxmaxmy); line(1); setoutput(3)
end;
V[8]=maxmaxmy;
format('maxmaxmx=-1.12345600-123?'); print(maxmaxmx);
if key(17) then format('maxmaxmy=-1.12345600-12300-wozytane?')
else format('maxmaxmy=-1.12345600-123?');
print(maxmaxmy);
format('maxmaxmz=-1.12345600-123?'); print(maxmaxmz); line(2);
405 DP;
y:=yp-dy; TODR(dpv,10,V); ITODR(dpiy,2,JZIY);
for i:=1 step 1 until iy do
begin
y:=y+dy; if key(22) then goto DALEJ; line(5);
format('w-APERTURZB-y/f=-12.1234?'); print(y);
print('lp z/f'); space(5); print('ksi theta'); space(6);
print('fi delta'); space(6); print('my m00 dB
space(6); print('mz m00 dB argz'); space(6);
print('m m dB alfa pol?');
FROMDR(dpkst,jz,KSI); FROMDR(dpth,jz,TH); FROMDR(dpf1,jz,FI);
FROMDR(dpd1,jz,DELTA); FROMDR(dpm,jz,MY); FROMDR(dpay,jz,AY);
FROMDR(dpmz,jz,MZ); FROMDR(dpez,jz,AZ); FROMDR(dpm,jz,M);
FROMDR(dpal,jz,ALFA); IFROMDR(dpst,jz,STER);
z:=zp-dz;
for j:=1 step 1 until jz do
begin
z:=z+dz;
403 if yy+z<x at RR then format('1200-1.1200') else format('1200-1.1200');

```

```

print(j, z);
ksl:=KSI[j]; theta:=TH[j]; f1:=FI[j];
delta:=DELTA[j]; my:=MY[j]/maxmaxmy;
lmy:=if my le c4 then -100.0 else c*ln(my);
argy:=if lmy=-100.0 then .0 else AY[j];
mz:=MZ[j]/maxmaxmy;
lmmz:=if mz le c4 then -100.0 else c*ln(mz);
argz:=if lmmz=-100.0 then 0 else AZ[j];
m:=M[j]; alfa:=ALFA[j]; ster:=STER[j];
lmm:=if abs(m) le c4 then -100.0 else c*ln(abs(m));
format(' 123.111123.111123.111123.1111234111');
print(ksl/grd, theta/grd, f1/grd, delta/grd, my);
print(lmy, argy/grd, mz, lmmz, argz/grd);
format(' 1.123111123.111123.111');
print(m, lmm, alfa/grd); STERO(ster); line(1);
450 end j;
if maxmaxmx le 10-5 then goto DALEJ; line(5);
print('lp'); space(7); print('mx mx dB
FROMDR(dpax,jz,MX); FROMDR(dpax,jz,AX);
format('12111.1234111123.111123.111123?');
z:=zp-dz;
for j:=1 step 1 until jz do
begin
z:=z+dz;
mx:=MX[j]/maxmaxmy;
lmmx:=if mx le c4 then -100.0 else c*ln(mx);
argx:=if lmmx=-100.0 then .0 else AX[j];
print(j, mx, lmmx, argx/grd, j);
end j;
DALEJ;
end i;
line(20);
end;
468 end?

```

```
4  comment APERTURA<< z dn 19.03.1980 cz II;
begin
integer dpmv, dpmz, dpay, dpaz, il, ilp,
dpm, dpal, dpv, dply, i, j, iy, jz, ster, k, K;
real p1, grd, dlq, dlq, ps, ps2, yp, dy, yk, y, zp, dz, zk, z, THRzer, krap,
Q, R, ksi, theta, fi, delta, mr, mrp, mp, mpp, THR, THP,
mod, faz, a1, a2, mx, my, mz, argx, argy, argz, maxmaxmy,
m, alfa, c4, c, lnmz, lnm, lnmz, odl, op, fz, prs,
RR, maxmaxmx, maxmaxmz, ro, chi, lnm, izop,
krizo, izok, izo, cx, cy, tx, ty;
integer array TEXT[1:30], JZLY[1:2];
array V[1:10];
boolean undef, discont, elips;
procedure TRANS5(chi, ro, y, z);
value chi, ro;
real chi, ro, y, z;
comment Oblicza wspolrzedne prostokatne (y, z)
punktu o wspolrzednych biegunowych chi(0, 2*pi),
ro(0, niesk);

begin
y:=ro*cos(chi); z:=ro*sin(chi)
end TRANS5;
23 procedure AXES(Y, Z, n);
integer n; array Y, Z;
comment Generuje biegunowy uklad wspolrzednych;
begin
integer m, i, j;
real grd, chi, ro;
grd:=arcsin(1.0)/90.0; m:=1;
Y[1]:=Z[1]=.0;
for i:=0 step 15 until 345 do
begin
chi:=i*grd; ro:=.0;
for j:=0 step 1 until 7 do
begin
ro:=ro+.5;
```

```

TRANS5(chi,ro,y,z);
if y le yk and y ge yp
and z le zk and z ge zp then
begin m:=m+1; Y[m]:=y; Z[m]:=z end;
end j;
end i;
n:=m;
end AXES;
45 procedure CARD(y,zp,dz,jz,izo,undef,discont,elips,F,Y,Z);
value y,zp,dz,jz,izo;
integer jz; real Y,zp,dz,izo;
boolean undef,discont,elips; array F,Y,Z;
comment Metoda interpolacji kwadratowej odszukuje
wspolrzedne z gory zadanych wartosci funkcji
znanej w wezlach rownoodleglej siatki punktow;
begin
integer mm,j1,j2,j3; boolean bin;
real mien,var,yy,z1,z2,z3,a1,a2,e3,p,q,s,w,W,Z1,Z2,p2;
mian:=2*dz*dz; yy:=y*y; var:=180.0;
mm:=jz div 2; if elips then izo:=izo+90.0;
for j:=1 step 1 until mm do
begin
if undef and abs(y) lt 10-5 then goto koniec;
j1:=2*j-1; j2:=j+1; j3:=j2+1;
z1:=zp+(j1-1)*dz; z2:=zp+j1*dz; z3:=zp+j2*dz;
if yy+z2*z2 gt RR then goto koniec;
if undef and abs(z2) lt 10-5 then goto koniec;
a1:=F[j1]; a2:=F[j2]; a3:=F[j3];
if elips then
begin
if abs(a1-360.0) lt 10-6 or abs(a2-360.0) lt 10-6
or abs(a3-360.0) lt 10-6
then goto koniec
end omi.janie trojek z punktami niekreslonymi;
if elips then
begin

```



```

var:=90.0;
a1:=a1+90.0; a2:=a2+90.0; a3:=a3+90.0;
end;
b1n:=abs(a1-a2) ge var or abs(a1-a3) ge var or abs(a2-a3). ge var;
if discont and b1n then
begin
if izo le var then
begin
if a1 gt var then a1:=a1-2*var;
if a2 gt var then a2:=a2-2*var;
if a3 gt var then a3:=a3-2*var;
end else
begin
if a1 le var then a1:=a1+2*var;
if a2 le var then a2:=a2+2*var;
if a3 le var then a3:=a3+2*var;
end;
end;
if discont then
begin
if abs(a1-a2)-var gt -v-6 or abs(a1-a3)-var gt -v-6
or abs(a2-a3)-var gt -v-6
then goto koniec
end
omijanie trojek z nieusuwalnymi skokami;
p:=a1-2.0*a2+a3; p2:=2.0*p;
if abs(p) gt .02244 then
begin
q:=(z2+z3)*a1+4.0*z2*a2-(z1+z2)*a3;
s:=z2*z3*a1-2.0*z1*z3*a2+z1*z2*a3-izo*mian;
w:=q-q-4.0*p*s;
if w lt .0 then goto koniec;
W:=sqrt(w);
Z1:=-((q+W)/p2; Z2:=-((q-W)/p2;
if abs(Z1-Z2) lt v-6 then goto dalej;
if Z1 lt z3 and Z1 ge z1 and yy+Z1*Z1 le RR then
begin

```

```
ll:=il+1;
if il gt ilp then begin setoutput(0); print('?il gt ilp?') end;
Y[il]:=y; Z[il]:=Z1
end;
//3 dalej;
if Z2 lt z3 and Z2 ge z1 and yy+Z2*Z2 le RR then
begin
  ll:=il+1;
  if il gt ilp then begin setoutput(0); print('?il gt ilp?') end;
  Y[il]:=y; Z[il]:=Z2
end;
goto koniec;
end else
  if abs(a1-a2) gt .02244 or abs(a2-a3) gt .02244
  or abs(a1-a3) gt .02244 then
    begin
      q:=a3-a1;
      s:=a1*x3-a3*x1-2*dz*xzo;
      Z1:=-s/q;
      if Z1 lt z3 and Z1 ge z1 and yy+Z1*Z1 le RR then
        begin
          ll:=il+1;
          if il gt ilp then begin setoutput(0); print('?il gt ilp?') end;
          Y[il]:=y; Z[il]:=Z1
        end;
        and else if not key(22) then
          begin
            format('?????y=-123.123????z:mod-1.1234-?do-1.1234?');
            if abs(a1*x3-a3*x1-2*dz*xzo) lt w-6
            or abs(a1*x3-a3*x1-2*dz*x(izo+360.0)) lt w-6
            then print(y, z1, z3);
          end;
        koniec;
      end j;
    end CARD;
  //4 procedure DRAW(X, Y, n, cx, cy, tx, ty);
```

```

value n, cx, cy, tx, ty;
integer n; real cx, cy, tx, ty; array l, i;
comment Perforuje tasme do plottera;
begin
integer i, f; real x, y, x0, y0, tx1, ty1;
setoutput(1); space(30); line(1); print('Q?');
x:=y:=0; format('x-012y-012f1?');
tx1:=if abs(tx) lt 999.0 then tx
else sign(tx)>999.0;
ty1:=if abs(ty) lt 999.0 then ty
else sign(ty)>999.0;
print(tx1, ty1, 2);
print('N? Q?');
for i:=1 step 1 until n do
begin
x0:=x; y0:=y; f:=1;
x:=x[i]>cx+tx; y:=y[i]>cy+ty;
if abs(x) gt 999.0 then xi:=sign(x)>999.0;
if abs(y) gt 999.0 then yi:=sign(y)>999.0;
if i=1 or abs(x-x0)+abs(y-y0) gt 500 then f:=2;
print(entier(xt.5), entier(yt.5), f);
end i;
print('x 999y 999f2?');
print('N'); space(30); line(1);
end DRAW;
/30 procedure OPIS(i1, izo, s, Y, Z);
integer i1; real izo; string s; array Y, Z;
comment Organizuje perforacje tasmy do plottera;
begin
if i1=0 then goto koniec;
setoutput(0); setinput(0);
format('=-123.12?'); print(s, izo);
DRAW(Y, Z, i1, cx, cy, tx, ty);
setoutput(3);
koniec;
end OPIS;
/31 procedure ZNAKI;

```

```
comment Generuje znaki rozdzielajace rodzin krzywych
begin
integer i; setoutput(1);
for i:=1 step 1 until 10 do print('abab');
end ZNAKI;

187 procedure FROMDR(dp,n,A);
integer dp,n; array A;
comment Sciaga z bebna od adresu dp
n elementowa real array A;

begin
drumplace:=dp; from drum(n,A[1]); dp:=drumplace
end FROMDR;

194 procedure IFROMDR(dp,n,A);
integer dp,n; integer array A;
comment Sciaga z bebna od adresu dp
n elementowa integer array A;

begin
drumplace:=dp; from drum(n,A[1]); dp:=drumplace
end IFROMDR;

201 pi:=2.0*arcsin(1.0); grd:=pi/180.0;
c1:=8.685889638; c4:=10-5;
dpy:=130000; FROMDR(dpv,10,V);
dpy:=130050; IFROMDR(dpy,2,JZLY);
yp:=V[1]; dy:=V[2]; yk:=V[3];
zp:=V[4]; dz:=V[5]; zk:=V[6];
maxmaxmx:=V[7]; maxmaxmy:=V[8]; maxmaxmz:=V[9];
R:=V[10]; RR:=R*R;
Jz:=JZLY[1]; Iy:=JZLY[2];
setinput(0); setoutput(0); instring(TEXT[1]);
setoutput(3); line(2); outstring(TEXT[1]);
line(1); format('1.123456',1237);
print('maxmaxmx='); print(maxmaxmx);
print('maxmaxmy='); print(maxmaxmy);
print('maxmaxmz='); print(maxmaxmz); line(1);
print('zmodyfikowany prom apert R/F=');
print(R); line(1);
```

```

1lp:=if 3*ly ge 200 then 3*ly else 200;
wait('k16-osie, k18-plot, k19-cala spert, k20-koniec izo?');
22/ if key(18) then
begin
  cxi=cy:=if key(19) then 220.0 else 440.0;
  txt=ty:=if key(19) then -200.0 else -400.0;
  format('=123.1,=123.1?');
  setoutput(3);
  print('?skalowanie: kx,ky'); print(cx,cy);
  print('przesumiecie: tx,ty'); print(tx,ty); line(1);
end;
begin
array MX,MY,MZ,AX,AY,AZ,FI,DELTA,
      KSI,TH,M,ALFA[1:jz],Y,Z[1:ilp];
if key(21) then print('zrodlo izotropowe??');
if key(18) then
begin
  if key(16) then
begin
  AXES(Y,Z,il);
  if not key(22) then
begin
line(5); print('uklad wspolrzecznych??');
print('lp'); space(7); print('y'); space(10); print('z??');
format('123=123.1234=123.1234?');
for i:=1 step 1 until il do print(i,Y[i],Z[i]);
end;
end;
setoutput(1); space(110); line(1);
wait('opisac tasme?'); print('PROGRAM APERTURA>>>?');
outstring(TEXT[1]); line(1);
$50if key(21) then print('zrodlo izotropowe??'); space(110);
if key(16) then
begin
wait('opisac tasme: osie?');
DRAW(Y,Z,il,cx,cy,tx,ty); space(110); ZNAKI;

```

```
end;
end;
line(5); setoutput(0); setinput(0);
print('izolinie my dB: izo od '); read(izo);
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then begin line(5); print('izolinie my dB') end;
K:=entier((izok-izo)/krizo)+1; izoi=izo-krizo;
wait('czy k20?');
for k:=1 step 1 until K do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpm:=10000;
264 if key(20) then goto konny;
if not key(22) then
begin
line(6); format('izo-my-dB=-123.123?'); print(izo);
end;
for i:=1 step 1 until iy do
begin
y:=y+dy;
FROMDR(dpm,y,jz,MY);
for j:=1 step 1 until jz do
begin
my:=MY[j]/maxmaxmy;
MY[j]:=if my le c4 then -100.0 else c*ln(my);
end j;
CARD(y, zp, dz, jz, izo, false, false, MY, Y, Z);
end i;
if not key(22) then
begin
format('.....123.....123.123456.....123.123456?');
for j:=1 step 1 until il do print(j, Y[j], Z[j]);
format(' ?ilp=123?'); print(il);
end;
if key(18) then OPIS(il, izo, 'my', Y, Z);
konny:
291 end k;
```

```
if key(18) then ZNAKI;
line(5); setoutput(0); setinput(0);
print('izolinie argy (stopnie): izo od '); read(izop);
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then
begin line(10); print('izolinie argy (stopnie)?') end;
k:=entier((izok-izop)/krizo)+1; izo:=izop-krizo;
wait('czy k20?');
for k:=1 step 1 until k do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpay:=40000;
if key(20) then foto konay;
if not key(22) then
begin
line(6); format('izo-ay=-123.123??'); print(izo)
end;
for i:=1 step 1 until iy do
begin
v:=y+dy;
FROMDR(dpay,jz,AY);
for j:=1 step 1 until jz do AY[j]:=AY[j]/grd;
CARD(y,zp,dz,jz,izo,false,true,false,AY,Y,Z);
end i;
if not key(22) then
begin
format('...123...-123. 123456...-123.123456?');
for j:=1 step 1 until il do print(j,Y[j],Z[j]);
format(' ?ilp=123?'); print(il);
end;
if key(18) then OPIS(il,izo,'argy',Y,Z);
end k;
if key(18) then ZNAKI;
line(5); setoutput(0); setinput(0);
print('izolinie mz dB: iz od '); read(izop);
```

304

323

```
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then
begin line(10); print('izolinie mz dB??') end;
K:=entier((izok-izop)/krizo)+1; izo:=izop-krizo;
wait('czy k20?');
for k:=1 step 1 until K do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpmz:=20000;
337 if key(20) then goto konmz;
if not key(22) then
begin
line(6); format('izo-mz-dB=-123.123??'); print(izo)
end;
for i:=1 step 1 until iy do
begin
y:=y+dy;
FROMDR(dpmz,jz,MZ);
for j:=1 step 1 until jz do
begin
mz:=MZ[j]/maxmzy;
MZ[j]:=if mz le o4 then -100.0 else o*ln(mz);
end j;
CARD(y, zp, dz, jz, izo, true, false, false, MZ, Y, Z);
end i;
if not key(22) then
begin
format('123.....123.123456.....123.123456?');
for j:=1 step 1 until il do print(j, Y[j], Z[j]);
format(' ?ilp=123? '); print(il);
end;
if key(18) then OPIS(il, izo, 'mz', Y, Z);
340 konmz;
end k;
if key(18) then ZNAKI;
line(5); setoutput(0); setinput(0);
```



```

print('izolinie argz (stopnie): izo od '); read(izop);
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then
begin line(10); print('izolinie argz ( stopnie )??') end;
K:=entier((izok-izop)/krizo)+1; izo:=izop-krizo;
wait('czy k20?');
for kt=1 step 1 until K do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpaz:=50000;
if key(20) then goto konaz;
if not key(22) then
begin line(6); format('izo-argz=-123.123??'); print(izo) end;
for i:=1 step 1 until iy do
begin
v:=v+dy;
FROMDR(dpaz,jz,AZ);
for j:=1 step 1 until jz do AZ[j]:=AZ[j]/grd;
CARD(y,zp,dz,jz,izo,krizo,true,false,AZ,Y,Z);
end i;
if not key(22) then
begin
format('.....123.....123.123456.....123.123456?');
for j:=1 step 1 until il do print(j,Y[j],Z[j]);
format(' ?ilp=123?'); print(il);
end;
if key(18) then OPIS(il,ize,'argz',Y,Z);
konaz:
end k;
if key(18) then ZNAKI;
line(5); setoutput(0); setinput(0);
print('izolinie m izo od '); read(izop);
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then
begin line(10); print('izolinie m??') end;
K:=entier((izok-izop)/krizo)+1; izo:=izop-krizo;

```

374

391

```

wait('czy k20?');
for k:=1 step 1 until K do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpm:=100000;
405 if key(20) then goto konm;
if not key(22) then
begin line(6); format('izo-m-dB--123.123??'); print(izo); end;
for i:=1 step 1 until iy do
begin
y:=y+dy;
FROMDR(dpm,jz,M);
CARD(y,zp,dz,jz,izo,false,false,M,Y,Z);
end i;
if not key(22) then
begin
format('.....123.....123.123456.....123.123456?');
for j:=1 step 1 until il do print(j,Y[j],Z[j]);
format('?ilp=123?'); print(il);
end;
if key(18) then OPIS(il,izo,'m',Y,Z);
42 konm;
end k;
if key(18) then ZNAKI;
line(5); setoutput(0); setinput(0);
print('izolinie alfa (stopnie): izo od '); read(izop);
print(' co '); read(krizo);
print(' do '); read(izok); line(1); setoutput(3);
if not key(22) then
begin line(10); print('izolinie alfa (stopnie)?') end;
K:=entier((izok-izop)/krizo)+1; izo:=izop-krizo;
wait('czy k20?');
for k:=1 step 1 until K do
begin
izo:=izo+krizo; y:=yp-dy; il:=0; dpal:=110000;
435 if key(20) then goto konal;
if not key(22) then

```

```

begin line(6); format('izo-alfa=-123.123??'); print(izo) end;
for i:=1 step 1 until iy do
begin
y:=y+dy;
FROMDR(dpal,jz,ALFA);
for j:=1 step 1 until jz do ALFA[j]:=ALFA[j]/grd;
CARD\y,zp,dz,jz,izo,false,true,true,ALFA,Y,Z);
end i;
if not key(22) then
begin
format('.....123.....123.123456.....123.123456?');
for j:=1 step 1 until il do print(j,Y[j],Z[j]);
format(' ?ilp=123?'); print(il);
end;
if key(18) then OPIS(il,izo,'alfa',Y,Z);
452 konal;
end k;
if key(18) then ZNAKI; setoutput(3);
line(5);
end;
454 end?

```