



ĆWICZENIE 3/4 ANALIZA ALGORYTMU SZYBKIEJ TRANSFORMATY FOURIERA (FFT I PROPAGACJA)

Wstęp teoretyczny

Transformata Fouriera.

Z punktu widzenia optyki transformacja Fouriera odpowiada rozkładowi frontu falowego na fale płaskie, propagujące się pod różnymi kątami w stosunku do głównej osi optycznej układu. Opis rozchodzenia się jednorodnej fali płaskiej w ośrodku jednorodnym i izotropowym nie jest złożony. Taka fala pozostaje nieskończoną, równomierną falą płaską, w której zmienia się tylko faza. Tak zdefiniowane podejście nie zawiera istotnych przybliżeń i – w zakresie skalarnej teorii dyfrakcji – uważane jest za niemal ścisłe. Taka interpretacja stanowi podstawę wielu algorytmów obliczeniowych i jest też podstawą tego ćwiczenia.

Transformację Fouriera stosują się także do rozkładu fali quasi-monochromatycznej na fale monochromatyczne.

Jednowymiarowe przekształcenie Fouriera.

Jednowymiarowa transformata (przekształcenie) Fouriera zdefiniowana jest przez następującą całkę:

$$\mathfrak{F}\{f(x)\} = F(\nu) = \int_{-\infty}^{+\infty} f(x) \exp[-2\pi i x \nu] dx \quad (1)$$

Funkcji f zmiennej x przyporządkowywana jest funkcja F zmiennej ν , zwanej częstością przestrzenną. Funkcja F zwana jest **transformatą Fouriera** funkcji f . F oraz f są w ogólności funkcjami o wartościach zespolonych.

Odwrotne przekształcenie Fouriera opisuje wzór:

$$\mathfrak{F}^{-1}\{F(\nu)\} = f(x) = \int_{-\infty}^{+\infty} F(\nu) \exp[2\pi i x \nu] d\nu \quad (2)$$



Aby funkcja posiadała transformatę Fouriera, powinna spełniać następujące warunki:

a. powinna być bezwzględnie całkowna w całej swej dziedzinie:

$$\int_{-\infty}^{+\infty} |f(x)| dx < \infty$$

b. powinna posiadać skończoną liczbę minimów i maksimów oraz nieciągłości na dowolnym, skończonym fragmencie dziedziny (zbioru liczb rzeczywistych)

c. nie powinna mieć nieciągłości nieskończonych.

Dwuwymiarowe przekształcenie Fouriera.

$$\mathfrak{S}\{f(x, y)\} = F(v_x, v_y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \exp[-2\pi i(xv_x + yv_y)] dx dy \quad (3)$$

Odwrotne dwuwymiarowe przekształcenie Fouriera określone jest następująco:

$$\mathfrak{S}^{-1}\{F(v_x, v_y)\} = f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(v_x, v_y) \exp[2\pi i(xv_x + yv_y)] dv_x dv_y \quad (4)$$

Znaczenie parzystości i nieparzystości funkcji.

Tabela 1 Parzystość i nieparzystość funkcji i jej transformaty Fouriera.

FUNKCJA	TRANSFORMATA
Rzeczywista i parzysta	Rzeczywista i parzysta
Rzeczywista i nieparzysta	Urojona i nieparzysta
Urojona i parzysta	Urojona i parzysta
Zespolona i parzysta	Zespolona i parzysta
Zespolona i nieparzysta	Zespolona i nieparzysta
Rzeczywista parzysta oraz urojona nieparzysta	Rzeczywista
Rzeczywista nieparzysta oraz urojona parzysta	Urojona



Przykłady często używanych funkcji i ich transformat.

Tabela 2 Często używane funkcje i ich transformaty Fouriera.

FUNKCJA	TRANSFORMATA
$rect(x, y) = rect(x) \cdot rect(y)$	$sinc(v_x) \cdot sinc(v_y)$
$circ(r) = circ(\sqrt{x^2 + y^2})$	$\frac{J_1(2\pi\rho)}{\rho}$, gdzie $\rho = \sqrt{v_x^2 + v_y^2}$
$\delta(x, y)$	1
$comb(x, y) = comb(x) \cdot comb(y)$	$comb(v_x) \cdot comb(v_y)$
$\exp[-\pi(x^2 + y^2)]$	$\exp[-\pi(v_x^2 + v_y^2)]$
$\exp[i\pi(x + y)]$	$\delta\left(v_x - \frac{1}{2}, v_y - \frac{1}{2}\right)$

Własności.

Tabela 3 Własności transformaty Fouriera.

FUNKCJA	TRANSFORMATA
$g(x)$	$G(v)$
$h(x)$	$H(v)$
$g(-x)$	$G(-v)$
$g^*(x)$	$G^*(-v)$
$g(ax)$	$\frac{1}{ a } G\left(\frac{v}{a}\right)$
$g\left(\frac{x-a}{b}\right)$	$ b \cdot e^{-2\pi i a v} \cdot G(bv)$
$\frac{d}{dx} g(x)$	$2\pi v \cdot G(v)$
$g(x) \cdot h(x)$	$G(v) \otimes H(v)$
$g(x) \otimes h(x)$	$G(v) \cdot H(v)$

Dyskretny szereg Fouriera.

Występujące w naturze wielkości fizyczne są opisane funkcjami tak różnorodnymi i skomplikowanymi, że nie jesteśmy w stanie wyznaczyć analitycznie ich transformat Fouriera, mimo że wiadomo, iż transformaty te istnieją. Koniecznym staje się numeryczne znajdowanie par transformat, a jest to możliwe dzięki dyskretnym przekształceniom Fouriera (prostemu



i odwrotnemu). Dyskretna Transformata Fouriera (DFT) operuje na funkcjach zdyskretyzowanych, a więc konieczne jest próbkowanie funkcji. Poniżej przedstawione rozważania dotyczą funkcji jednowymiarowych.

Jeżeli rozważamy ciąg okresowy $x(n)$ o okresie N , to możliwe jest jego przedstawienie za pomocą szeregu Fouriera, mającego postać sumy zespolonych wyrazów wykładniczych, w których częstotliwości są wielokrotnościami częstości podstawowej $2\pi/N$ ciągu okresowego.

Szereg Fouriera ciągu okresowego, w przeciwieństwie do szeregu Fouriera okresowej funkcji ciągłej, ma tylko N różnych zespolonych wyrazów wykładniczych, z których każdy ma okres będący podwielokrotnością podstawowego okresu N . Zatem okresowy ciąg $x(n)$ można przedstawić w postaci szeregu Fouriera zawierającego tylko N wyrazów:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left(2\pi i \frac{nk}{N}\right) \quad (5)$$

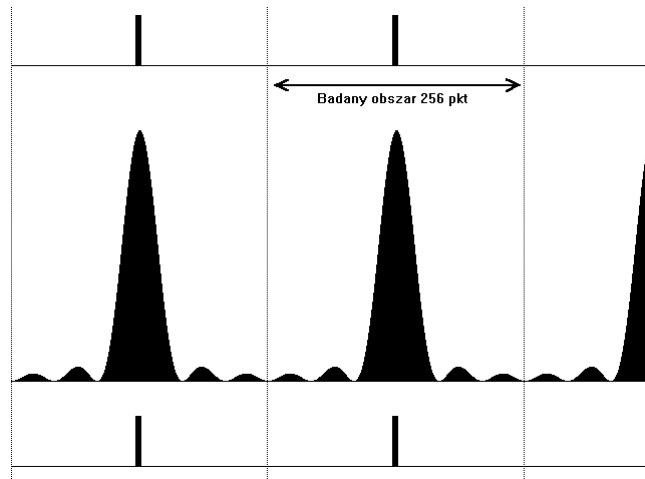
Istnienie współczynnika $1/N$ wiąże się z koniecznością normalizacji. Współczynniki $X(k)$ określone są zależnością:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp\left(-2\pi i \frac{nk}{N}\right) \quad (6)$$

Przedstawianie ciągów okresowych za pomocą dyskretnego szeregu Fouriera można wykorzystać także do fourierowskiego przedstawiania ciągów o skończonej długości; jest ono wtedy nazywane dyskretną transformatą Fouriera. Wymaga ono jednakże odpowiedniej interpretacji: konieczne jest przedstawienie ciągu o skończonej długości N za pomocą ciągu okresowego o okresie N , który w ramach jednego okresu jest identyczny z ciągiem o skończonej długości. Ponieważ ciąg okresowy można przedstawić jednoznacznie za pomocą dyskretnego szeregu Fouriera, to można w ten sposób, również jednoznacznie, przedstawić wyjściowy ciąg o skończonej długości (mając dyskretny szereg Fouriera można odtworzyć ciąg okresowy, a następnie, biorąc jeden okres, uzyskać ciąg wyjściowy).

Praktyczne wykorzystanie DFT.

Zerowy rząd widma znajduje się w elemencie $X(0)$ ciągu wyjściowego, a cały ciąg reprezentuje jedynie jeden okres widma. Okresowy ciąg wejściowy $x(n)$ (dla n całkowitych) można traktować jako zbiór próbek okresowej funkcji ciągłej. Widmo fourierowskie takiej funkcji jest okresowe (bo funkcja była próbkowana) i dyskretne (bo funkcja była okresowa). Wykonanie odwrotnej transformaty Fouriera takiego widma da funkcję okresową i dyskretną.



Rysunek 1 Widmo Fouriera funkcji próbkowanej i okresowej (jest okresowe i dyskretne).

Jeżeli badana funkcja osiąga niezerowe wartości na krańcach okresu, to często obserwuje się zniekształcenia widma wskutek powstawania nieciągłości (są one źródłem niepożądanych harmonicznych).

Szybka transformata Fouriera (FFT).

Na możliwość znacznej redukcji liczby obliczeń zwrócono uwagę dopiero w roku 1965, gdy Cooley i Tukey opublikowali algorytm wyznaczania dyskretnej transformaty Fouriera dla ciągów, których długość N jest liczbą złożoną (jest iloczynem dwóch lub więcej liczb całkowitych).

Algorytmy FFT wykorzystują własności symetrii i okresowości odpowiednich funkcji. Algorytmy te zastępują obliczenia dyskretnej transformaty Fouriera dla ciągu o długości N , przez obliczenia dla ciągów odpowiednio krótszych.

Algorytm FFT z podziałem czasowym.

Algorytm FFT jest najefektywniejszy i ma najprostszą postać dla ciągów o długości N , będącej wielokrotnością 2, tzn. 2^p , gdzie p jest liczbą naturalną. Zastosowanie tego algorytmu umożliwia zmniejszenie liczby działań w trakcie obliczania dyskretnej transformaty Fouriera do $(N/2)\log_2 N$ mnożeń i $N\log_2 N$ sumowań liczb zespolonych, w stosunku do koniecznych N^2 mnożeń i $N(N-1)$ sumowań zespolonych w przypadku korzystania bezpośrednio z definicji DFT.

Aby osiągnąć wzrost wydajności obliczeń, należy rozłożyć obliczenia dyskretnej transformaty Fouriera na obliczenia o stopniowo zmniejszającym się rozmiarze.

Przykładowo oblicza się dwie dyskretne transformaty Fouriera o okresie $N/2$, a następnie odpowiednio się je łączy. Wyznaczenie tych transformat oraz operacja ich połączenia wymagają mniejszej liczby działań na liczbach zespolonych niż bezpośrednie obliczenie ciągu.

Obliczenia sprowadzają się więc do wyznaczenia wielu transformat dwupunktowych, a następnie odpowiedniego ich łączenia w coraz większe transformaty, aż do uzyskania jednej, N -punktowej transformaty.

Całe to postępowanie owocuje, jak to wcześniej zaznaczono, znacznym ograniczeniem liczby działań arytmetycznych, dzięki unikaniu wielokrotnego obliczania tych samych danych



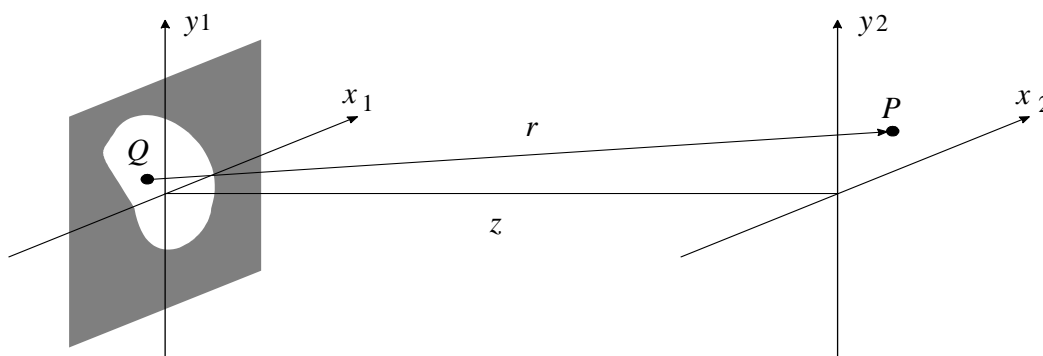
(nastąpiła znaczna optymalizacja procesu obliczeń względem klasycznego podejścia do wyznaczania DFT).

Można wprowadzić następującą interpretację tej operacji. Wykonywane są dwie (zamiast jednej) dyskretne transformaty Fouriera badanej funkcji, ale dla próbkowania dwa razy rzadszego niż jest to pożądane; jednakże informacja nie jest tracona, ponieważ próbki pobierane były w obu przypadkach z różnych punktów i informacje zawarte w obu transformatach wzajemnie się uzupełniają.

Obliczenie 2-wymiarowej transformaty Fouriera na macierzy złożonej z m wektorów o długości n sprowadza się do policzenia m transformat (długości n) po wierszach. Następnie obliczamy n transformat (długości m) po kolumnach.

Przykładowy kod FFT załączony jest w Załączniku A.

Numeryczna propogacja.



Rysunek 2 Schemat płaszczyzn w rozważanym układzie optycznym.

W celu obliczenia zespolonej amplitudy pola świetlnego u_2 w odległości z od rozkładu wejściowego u_1 należy obliczyć poniższą całkę. Zakładamy parametry typowe dla strefy Fresnela.

$$u_2(x_2, y_2) = \frac{\exp(ikz)}{i\lambda z} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u_1(x_1, y_1) \exp\left[\frac{ik}{2z} \left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right]\right] dx_1 dy_1 \quad (7)$$

Obliczenia mogą zostać dokonane różnymi sposobami.

Bezpośrednia metoda.

Polega na bezpośrednim liczeniu numerycznym całki (7). Istnieje tu jednak poważny problem. Parametr z występujący w czynniku Fresnela jest umieszczony w mianowniku ułamka. Stąd przy małej wartości parametru z występują duże zmiany fazy, co z kolei powoduje szybkie oscylacje. Niezbędna jest duża ilość punktów próbkowania w płaszczyźnie x_1, y_1 .

We współrzędnych kartezjańskich całkę (7) można przepisać do następującej postaci.



$$u_2(x_2, y_2) = \frac{\exp(ikz)}{i\lambda z} \cdot \exp\left[\frac{ik}{2z}(x_2^2 + y_2^2)\right] \cdot \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} u_1(x_1, y_1) \cdot \exp\left[\frac{ik}{2z}(x_1^2 + y_1^2)\right] \cdot \exp\left[-\frac{2\pi i}{z\lambda}(x_1 \cdot x_2 + y_1 \cdot y_2)\right] dx_1 dy_1 \quad (8)$$

Całka może być liczona przy użyciu standardowych metod numerycznych (np. metodą prostokątów, trapezów itp.). Należy pamiętać, że obliczane wielkości są liczbami zespolonymi. Zaletą metody jest łatwość przeskalowywania współrzędnych x_1, y_1 oraz x_2, y_2 . Wadą jest bardzo długi czas obliczeń.

Metoda FFT.

Metoda bazuje na fakcie, że zespoloną amplitudę w płaszczyźnie wyjściowej można przedstawić w postaci transformacji Fouriera pola wejściowego pomnożonego przez czynnik fresnelowski.

$$u_2(x_2, y_2) = \frac{\exp(ikz)}{i\lambda z} \exp\left[\frac{ik}{2z}(x_2^2 + y_2^2)\right] \mathfrak{F}\left\{u_1(x_1, y_1) \exp\left[\frac{ik}{2z}(x_1^2 + y_1^2)\right]\right\}_{|v_x, v_y} \quad (9)$$

gdzie $\mathfrak{F}\{..\}$ oznacza transformację Fouriera.

Podstawową zaletą metody jest szybkość obliczeń. Często za wadę metody uważa się, że współrzędne wejściowe (x_1, y_1) i wyjściowe (x_2, y_2) są przeskalowane przez czynnik, który zależy od odległości z . Płaszczyzna wyjściowa powinna być próbkowana z inną częstotliwością niż płaszczyzna wejściowa.

Istotnym mankamentem jest fakt, że parametr z w czynniku Fresnela jest umieszczony w mianowniku ułamka. Małe zmiany parametru z powodują duże zmiany fazy, co z kolei powoduje szybkie oscylacje. Niezbędna jest duża ilość punktów próbkowania w płaszczyźnie x_1, y_1 .

SPLIT – dwukrotne użycie FFT w celu obliczenia splotu.

Metoda bazuje na rozkładzie pola wejściowego na fale płaskie. Warto wspomnieć, że jest to bardziej ogólne podejście do zagadnienia dyfrakcji niż całka (7). Jest to praktycznie zupełnie ścisła metoda w ramach skalarnej teorii dyfrakcji.

$$u_2(x, y) = u_1(x, y) \otimes h(x, y; \lambda, z); h(x, y; \lambda, z) = \frac{\exp(ikz)}{i\lambda z} \exp\left[\frac{ik}{2z}(x^2 + y^2)\right] \quad (10)$$

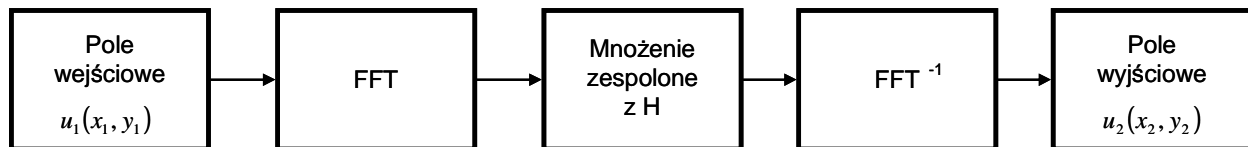
Rozkład wyjściowy można przedstawić jako splot rozkładu wejściowego z funkcją odpowiedzi impulsowej swobodnej przestrzeni.

$$H(v_x, v_y) = \mathfrak{F}\{h(x, y)\} = \exp(ikz) \cdot \exp[-i\pi\lambda z(v_x^2 + v_y^2)] \quad (11)$$



Istnieje również następująca interpretacja. Rozkład wejściowy jest rozkładany na fale płaskie o różnych amplitudach i kątach propagacji. Każda fala płaska - niezależnie od innych - doznaje stosownego opóźnienia fazowego. Rozkład wyjściowy jest superpozycją "skorygowanych" fal płaskich.

W celu łatwego i szybkiego obliczenia splotu korzysta się z procedury FFT - jako narzędzia metod numerycznych.



Rysunek 3 Algorytm obliczania splotu.

Do zalet metody można zaliczyć dwa fakty. Wejściowe (x_1, y_1) i wyjściowe (x_2, y_2) współrzędne są identyczne. Faza funkcji przenoszenia $H(v_x, v_y)$ liniowo zależy od z .

Zadania do wykonania:

Przeprowadzenie symulacji w aplikacji LightSword (dostępna w laboratorium) na macierzy 2048 x 2048 punktów. Próbkowanie 10 μm x 10 μm . Długość fali światła $\lambda=0.6328 \mu\text{m}$

- 1) Wygenerować dwuwymiarową funkcję stałą. Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy. Dokonać ponownej transformaty Fouriera. Zarejestrować rozkłady amplitudy i fazy.
- 2) Wygenerować dwuwymiarową funkcję rectus umieszczoną centralnie (małe rozmiary – 10 x 10 pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 3) Wygenerować dwuwymiarową funkcję rectus umieszczoną z przesunięciem (małe rozmiary – 10 x 10 pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1. Sprawdzić twierdzenie o przesunięciu.
- 4) Wygenerować dwuwymiarową funkcję rectus (z losową fazą) umieszczoną centralnie (małe rozmiary – 10 x 10 pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 5) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (małe rozmiary – promień 10 pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.



- 6) Wygenerować dwuwymiarową funkcję circus (z losową fazą) umieszczoną centralnie (małe rozmiary – promień 10 pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 7) Wygenerować dwuwymiarową funkcję rectus umieszczoną centralnie (duże rozmiary – kilkaset na kilkaset pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 8) Wygenerować dwuwymiarową funkcję rectus (z losową fazą) umieszczoną centralnie (duże rozmiary – kilkaset na kilkaset pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 9) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (duże rozmiary – promień kilkaset pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 10) Wygenerować dwuwymiarową funkcję circus (z losową fazą) umieszczoną centralnie (duże rozmiary – promień kilkaset pikseli). Obliczyć widmo Fouriera. Zarejestrować rozkłady amplitudy i fazy oraz przekrój poprzeczny w trybie logarytmicznym z powiększeniem 1:1.
- 11) Obliczyć splot 2 dwuwymiarowych funkcji rectus o identycznych rozmiarach (100 x 100 pikseli). Zarejestrować wynikowy rozkład amplitudy oraz przekrój poprzeczny.
- 12) Obliczyć splot 2 dwuwymiarowych funkcji rectus o różnych rozmiarach (100 x 100 pikseli i 50 x 50 pikseli). Zarejestrować wynikowy rozkład amplitudy oraz przekrój poprzeczny.
- 13) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality high, Double window) na odległość $z=1580\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 13.TAB).
- 14) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality normal, NoWindow) na odległość $z=1580\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 14.TAB).
- 15) Odjąć od pliku 14.TAB plik 13.TAB. Zarejestrować rozkład natężenia i zinterpretować wynik.
- 16) Wygenerować dwuwymiarową funkcję circus styczną do brzegu analizowanego obszaru (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality high, Double window) na odległość $z=1580\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 16.TAB).



- 17) Wygenerować dwuwymiarową funkcję circus styczną do brzegu analizowanego obszaru (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality normal, NoWindow) na odległość $z=1580\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 17.TAB).
- 18) Odjąć od pliku 17.TAB plik 16.TAB. Zarejestrować rozkład natężenia i zinterpretować wynik.
- 19) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality high, Double window) na odległość $z=158\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 19.TAB).
- 20) Wygenerować dwuwymiarową funkcję circus umieszczoną centralnie (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality normal, NoWindow) na odległość $z=158\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 20.TAB).
- 21) Odjąć od pliku 20.TAB plik 19.TAB. Zarejestrować rozkład natężenia i zinterpretować wynik.
- 22) Wygenerować dwuwymiarową funkcję circus styczną do brzegu analizowanego obszaru (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality high, DbWindow) na odległość $z=158\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 22.TAB).
- 23) Wygenerować dwuwymiarową funkcję circus styczną do brzegu analizowanego obszaru (promień $R=1\text{mm}$). Obliczyć propagację (on-axis, quality normal, NoWindow) na odległość $z=158\text{ mm}$. Zarejestrować rozkład natężenia (zapisać jako 23.TAB).
- 24) Odjąć od pliku 23.TAB plik 22.TAB. Zarejestrować rozkład natężenia i zinterpretować wynik.
- 25) Przeprowadzić symulację obrazowania obiektu dwuwymiarowego w układzie 2f-2f. Sprawdzić rozkład pola poza płaszczyzną obrazowania (np. 10mm). Przyjąć f około 200 mm .
- 26) Przeprowadzić symulację obrazowania obiektu dwuwymiarowego z rozpraszaczem w układzie 2f-2f. Sprawdzić rozkład pola poza odległością obrazowania (np. 10 mm). Przyjąć f około 200 mm .
- 27) Wykonać górnoprzepustową i dolnoprzepustową filtrację przestrzenną na binarnym obiekcie graficznym (przy użyciu funkcji Band Pass Filter - Amplitude). Zarejestrować wynikowe rozkłady amplitudy.
- 28) Wykonać górnoprzepustową i dolnoprzepustową filtrację przestrzenną na szaroodcieniowym obiekcie graficznym (przy użyciu funkcji Band Pass Filter - Amplitude). Zarejestrować wynikowe rozkłady amplitudy.
- 29) Uzyskać na obrazie quasi periodycznym – poprzez filtrację przestrzenną – efekt odwrócenia kontrastu.



- 30) Odfiltrować periodyczne zaburzenie z obrazu poprzez filtrację przestrzenną.
- 31) Odfiltrować obraz z tego samego zdjęcia (pozostawiając tylko periodyczne zaburzenie).
- 32) Dokonać korelacji binarnego obrazu przedstawiającego element elektroniczny z binarnym obrazem przedstawiającym obwód drukowany. Zarejestrować rozkład natężenia. Zidentyfikować piki korelacyjne. Zarejestrować ich przekroje poprzeczne.
- 33) Dokonać korelacji szaroodcieniowego obrazu przedstawiającego element elektroniczny z szaroodcieniowym obrazem przedstawiającym obwód drukowany. Zarejestrować rozkład natężenia. Zidentyfikować piki korelacyjne. Zarejestrować ich przekroje poprzeczne.
- 34) Symulacja obrazowania w świetle niespójnym (analogia do wirującej matówki). Najpierw przeprowadzić symulację obrazowania obiektu dwuwymiarowego z rozpraszaczem w układzie 2f-2f (zapisać jako obraz01.TAB). Następnie wykonać 9 takich symulacji za każdym razem przesuwając dyfuzor o kolejny 1mm (przy użyciu funkcji Roll), zapisując je jako kolejne pliki TAB. Na końcu dodać wszystkie pliki TAB. Zestawić obrazy uzyskane dla kolejnych sum i porównać wyniki. Sformułować wnioski dotyczące ilości szumu w uzyskanych obrazach.
- 35) Symulacja propagacji frontu falowego za przezroczem i za amplitudową siatką binarną (ugięcie na amplitudowej siatce dyfrakcyjnej), w świetle koherentnym dla różnych długości fali światła (RGB - 632.8nm, 532nm, 428nm).



Załącznik A

Poniżej przedstawiona została przykładowa procedura napisana przez mgr inż. Mariusza Krukowskiego.

```
/* ***** START Inicjalizacja FFT START ***** */
```

```
void fftinit(unsigned int N,unsigned int p)
```

```
{
```

```
    PiN=2*3.141592654/N;
```

```
    tabl[0]=0;
```

```
    m=1;
```

```
    for(i=1;i<=p;i++)
```

```
    {
```

```
        m2=m*2;
```

```
        mm=N/m2;
```

```
        for(j=0;j<m;j++) tabl[j+m]=tabl[j]+mm;
```

```
        m*=2;
```

```
    }
```

```
    j=N>>1;
```

```
    for(i=0;i<j;i++)
```

```
    {
```

```
        sinus[i]=sin(PiN*i);
```

```
        cosinus[i]=cos(PiN*i);
```

```
    }
```

```
}
```

```
/* ***** START FFT START ***** */
```

```
unsigned int ip,ni,nj,ii,iii,N2;
```

```
unsigned int adr1,adr2,adr3,phi;
```

```
float re,im;
```

```
void fft(unsigned int N, unsigned int p)
```

```
{
```

```
    for(i=0;i<N;i++)
```

```
    {
```

```
        Bre[i]=Are[tabl[i]];
```

```
        Bim[i]=Aim[tabl[i]];
```

```
    }
```

```
    ii=1;
```

```
    iii=N;
```

```
    for(ip=1;ip<=p;ip++)
```

```
    {
```

```
        ii<<=1;
```

```
        iii>>=1;
```

```
        N2=N/ii;
```

```
        for(ni=0;ni<N2;ni++)
```

```
        {
```

```
            adr2=ii>>1;
```



```
adr1=ni*ii;
phi=0;
for(nj=0;nj<adr2;nj++)
{
  adr3=adr1+adr2;

  re=Bre[adr3]*cosinus[phi] + Bim[adr3]*sinus[phi];
  im=Bim[adr3]*cosinus[phi] - Bre[adr3]*sinus[phi];

  Bre[adr3]=Bre[adr1] - re;

  Bim[adr3]=Bim[adr1] - im;

  Bre[adr1] += re;

  Bim[adr1] += im;

  adr1++;
  phi+=iii;
}
}

/* dla odwrotnej podzielic wszystkie elementy przez N */
}
```