Zadania do Metod Numerycznych II A
Grupy 2, 3. Seria 3.

Dyskretna transformata Fouriera i jej zastosowania

Jakub Narębski

25.05.2001 r.

1 Obliczenie dyskretnej transformaty Fouriera z definicji

Napisać program obliczający zespoloną dyskretną transformatę Fouriera z jej definicji:

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\pi \imath kn/N} \tag{1}$$

gdzie $\imath$ oznacza jednostkę urojoną (t.j. ze wzoru (15) w `four.ps` albo (1a) w `fft.ps`). Sprawdzić, czy odwrotna transformata Fouriera, zadana wzorem:

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi \imath kn/N} \tag{2}$$

przeprowadza otrzymaną tablicę H z powrotem w oryginalną funkcję.

Można przykładowo wziąć funkcję zadaną wzorem (wypełnić tablicę h w następujący sposób):

$$h_k = \sin(2\pi k/N) + \imath\, 0$$
$$h_k = \exp(\imath 2\pi 8k/N)$$
$$h_k = \begin{cases} 1 & \text{dla } k < N/2 \\ \imath & \text{dla } k \geq N/2 \end{cases}$$

Zrobić wykres oryginalnej funkcji, jej transformaty Fouriera oraz odwrotnej transformaty Fouriera transformaty Fouriera. Narysować część rzeczywistą i urojoną lub moduł (wartość bezwględną). Pod Linuksem można to zrobić za pomocą programu `gnuplot` poleceniem (zakładając, że do pliku *h.dat* zostały zapisane kolejne elementy tablicy h, w każdym wierszu część rzeczywista i część urojona (oddzielone np. spacją)):

```
plot 'h.dat' using 1 title "re(h)" with linespoints
replot 'h.dat' using 2 title "im(h)" with linespoints
replot 'h.dat' using (abs({$1,$2})) title "abs(h)" with
linespoints
```

i analogicznie dla transformaty Fouriera (zapisanej np. w pliku *H.dat* lub jako następne kolumny jednego pliku z danymi), oraz transformaty odwrotnej.

Polecenie `replot` powoduje wykreślenie dodatkowych danych na bieżącym wykresie. Można pominąć fragment `with linespoints`: służy on do określenia sposobu wykreślania danych (tutaj: punkty połączone liniami). `{a,b}` oznacza w gnuplocie liczbę zespoloną $a + \imath b$. Argument `using` z następującym po nim wyrażeniem określa jakie dane będą wykreślone na wykresie. Patrz także `help plot datafile` w gnuplocie.

2 Właściwości transformaty Fouriera

Napisać program sprawdzający własności transformaty Fouriera. Sprawdzić jakie własności ma transformata Fouriera funkcji

- rzeczywistej, parzystej
- rzeczywistej, nieparzystej
- urojonej, parzystej
- urojonej, nieparzystej

gdzie *nieparzystość* oznacza, że $h_k = h_{N-k} \equiv h_{-k}$.

Uzupełnienie: sprawdzić własność tzw. *aliasingu*, t.j. wykreślić funkcję o częstości większej od **krytycznej częstości Nyquista** $2/N$, jej wartości w punktach próbkowania, jej dyskretną transformatę Fouriera oraz transformatę odwrotną.

Przykładową funkcją może być $h(t) = sin(2\pi\frac{3}{2}Nt)$, gdzie $t_k = k/N$, zaś $h_k = h(t_k)$.

3 Filtrowanie przy użyciu transformaty Fouriera

Zrobić prosty filtr przy użyciu dyskretnej transformaty Fouriera, tzn. obliczyć transformatę Fouriera, wyciąć odpowiednie częstości (pamietając o tym, że dla zwykłej, zespolonej dyskretnej transformaty Fouriera po indeksie $N/2$ następują częstości ujemne, t.j. $H_{N-n} \equiv H_{-n}$ dla $n > \frac{N}{2}$). Przykładem może być prosty filtr pasmowy: wycinamy nieporządane częstości. Mogą być usuwane częstości wysokie jeśli wiemy, że sygnał badany jest wolnozmienny; częstości niskie jeśli chcemy usunąć wolnozmienny trend. Możemy też sostawić tylko częstości w pewnym zakresie wokół danej częstości, jeżeli sygnał jest sygnałem o modulowanej amplitudzie lub częstości).

Uzupełnienie: zrobić to także przy użyciu rzeczywistej transformaty Fouriera, wyliczonej za pomocą własnej procedury (na podstawie `fourrsc.ps`) lub procedury `realft` z „Numerical Recipies in C" (zawartej w pliku `fft_NRC.tar`).

Jakub Narębski