

Transformata Fourier folosind MATLAB

1.1. Transformata Fourier pentru semnale continue în timp

În prima lucrare s-a discutat despre tehnici de analiză a semnalelor periodice în timp folosind funcții de tip cosinus sau sinus pentru a exprima semnalele periodice ca o combinație liniară a acestor funcții. Însă, majoritatea semnalelor cu care lucrăm sunt neperiodice. De aceea, pentru a analiza spectrul acestor semnale, în domeniul frecvență, se folosește Transformata Fourier.

Având un semnal $x(t)$ neperiodic ne întrebăm în ce domeniul de frecvență există acest semnal, sau, altfel spus, care este combinația de frecvențe necesare pentru a reconstrui semnalul folosind doar funcții standard de tip cosinus. Știind deja cum se reprezintă un semnal periodic, ca suma de funcții standard de tip cosinus, se va construi din semnalul neperiodic un semnal periodic prin repetarea semnalului neperiodic cu o perioadă T . Folosind acest procedeu se va obține în final că transformata Fourier a unui semnal neperiodic $x(t)$ este:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (1)$$

Transformata Fourier, este o funcție continuă de frecvență. Dacă știm expresia funcției $X(\omega)$, pentru a reconstrui semnalul inițial $x(t)$ vom folosi transformata Fourier inversă folosind formula:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega \quad (2)$$

Condițiile de existență a transformatei Fourier sunt:

- semnalul $x(t)$ trebuie să fie integrabil în sens absolut;
- dacă $x(t)$ este un semnal discontinuu, atunci trebuie să aibă un număr finit de discontinuități pe orice interval de timp;
- semnalul $x(t)$ trebuie să aibă un număr finit de minime și maxime în orice interval de timp.

Pentru diferitele semnale uzuale, o lista a transformatelor Fourier¹ poate fi găsită aici <http://www.thefouriertransform.com/pairs/fourier.php>.

¹ <https://allsignalprocessing.com/2019/10/22/fourier-methods-prominent/>

1.2. Transformata Fourier discretă

Având în vedere că transformata Fourier este rezultatul unei operații matematice de tip integrare care nu poate fi rezolvată analitic de un sistem de calcul, va trebui să folosim diferite metode numerice pentru a o calcula. Astfel, pentru a calcula transformata Fourier folosind MATLAB ne vom folosi de ceea ce se numește Transformata Fourier Discretă, care este o aproximare a transformatei Fourier.

Transformata Fourier în timp discret a unei secvențe $x[n] = x(nT_s)$ este dată de relația:

$$X\left(\frac{kF_s}{N}\right) \cong T_s \sum_{n=0}^N x(nT_s) e^{-j2\pi kn/N} \quad (3)$$

unde

- $T_s = \frac{1}{F_s}$ este perioada de eșantionare a semnalului $x(t)$, aleasă astfel încât semnalul să nu aibă variații foarte mari în intervalul de timp T_s ;
- k este indicele de sumare;
- N reprezintă numărul de puncte de eșantionare, astfel încât durata semnalului să fie $N \cdot T_s$. Cu cât N este mai mare, cu atât aproximarea spectrului de frecvențe continue este mai bună.

În MATLAB, pentru a calcula transformata Fourier Discretă, se folosește funcția **fft()**. Denumirea sa reprezintă prescurtarea de la *Fast Fourier Transform* (transformata Fourier rapidă) și indică faptul că este folosită pentru a calcula un algoritm rapid. Funcția **fft()** este descrisă în MATLAB aici:

<https://www.mathworks.com/help/MATLAB/ref/fft.html>

$X = \text{fft}(x, N)$

- X conține valorile transformatei Fourier discrete aplicată elementelor vectorului x ;
- lungimea transformatei Fourier de lungime N .

Intervalul de referință pentru reprezentarea în frecvență este $[0, F_s)$. De obicei se dorește însă reprezentarea în intervalul de bază $[-F_s/2, F_s/2)$. Având în vedere faptul că transformata Fourier discretă este periodică de perioadă F_s , reprezentarea din intervalul $[-F_s/2, 0)$ corespunde cu reprezentarea din intervalul $[F_s/2, F_s)$. Prin urmare trebuie realizată o inversare a celor două jumătăți ale vectorului X , prin utilizarea funcției **fftshift()**.

Funcția **fftshift()** este descrisă în MATLAB aici:

<https://www.mathworks.com/help/matlab/ref/fftshift.html>

`Xshift = fftshift(X)`

- **Xshift** este vectorul obținut prin schimbarea celor două jumătăți ale vectorului **X** între ele;

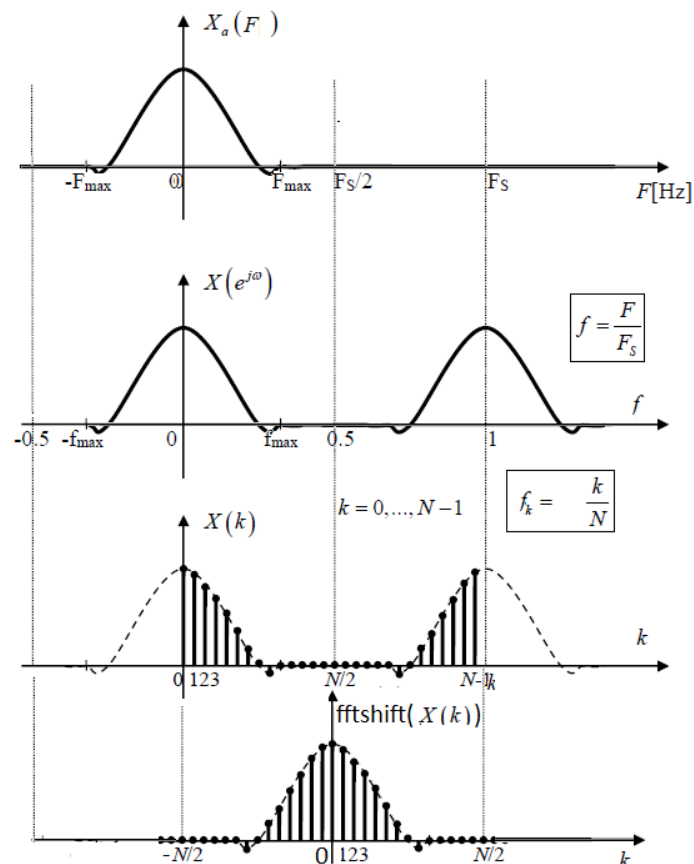


Fig. 1. Reprezentările spectrului unui semnal discret în funcție de frecvența normală și corespondența cu frecvența analogică.

Correspondența între componentele spectrale de indice k calculate cu TFD și spectrul reprezentat în frecvențe normale.

Exemplu 1

Folosind MATLAB calculați și reprezentați modulul și faza Transformatei Fourier Discrete a secvenței:

$$x(t) = \begin{cases} t(1-t) & , 0 \leq t < 1 \\ 0 & , 1 \leq t < 2 \end{cases}$$

% Calculul transformatei Fourier folosind 32 de esantioane in intervalul %% 0-2 secunde

```
close all;
clear
clc
```

```

Fs = 16; % frecventa de esantionare
Ts = 1/Fs; % perioada de esantionare
N = 512; % Lungimea transformatei Fourier
df = Fs/N; % rezolutia in domeniul frecventa
Tmax = 2; % durata semnalului
% vectorul valorilor timpului la momentele de esantionare
t = 0:Ts:Tmax-Ts;
% vectorul valorilor functiei in momentele de esantionare
x = t.*(1-t).*(sign(t)-sign(t-1))/2;

X = Ts*fft(x,N); % transformata Fourieri discret
F = 0:df:Fs/2-df; % valorile indicelui frecventelor

% Xshift = Ts*fftshift(fft(x,N));
% Fshift = -Fs/2:df:Fs/2-df;

%Graficele
subplot(3,1,1);
p = plot(t,x,'k');
set(p,'LineWidth',2); grid on;
xlabel('Timpul, t(s)');
ylabel('x(t)');
subplot(3,1,2);
p = plot(F,abs(X(1:N/2)),'k');
% p = plot(Fshift,abs(Xshift),'k');
set(p, 'LineWidth',2); grid on;

xlabel('Frecventa, F(Hz)');
ylabel('IX(f)I')
subplot(3,1,3);
p = plot(F,angle(X(1:N/2)),'k');
% p = plot(Fshift,angle(Xshift),'k');
set(p,'LineWidth',2); grid on;

xlabel('Frecventa, F(Hz)');
ylabel('Faza X(f)');

```

După executarea codului se obține Fig. 2, care conține reprezentarea în timp a semnalului $x(t)$ și reprezentarea modulului și fazei Transformatei Fourier Discrete $X(f)$, calculate cu ajutorul funcției `fft()`.

Prin comentarea liniilor 15, 16, 28, 35:

- $X = Ts*fft(x,N);$

- $F = 0:df:Fs/2-df;$
- $p = \text{plot}(F, \text{abs}(X(1:N/2)), 'k');$
- $p = \text{plot}(F, \text{abs}(X(1:N/2)), 'k');$

și de-comentarea liniilor 18, 19, 29, 36:

- $X_{\text{shift}} = Ts * \text{fftshift}(\text{fft}(x, N));$
- $F_{\text{shift}} = -Fs/2:df:Fs/2-df;$
- $p = \text{plot}(F_{\text{shift}}, \text{abs}(X_{\text{shift}}), 'k');$
- $p = \text{plot}(F_{\text{shift}}, \text{angle}(X_{\text{shift}}), 'k');$

se obțin modulul și faza Transformatei Fourier Discrete $X(f)$ după inversarea celor două jumătăți ale vectorului X cu ajutorul funcției `fftshift()`. Graficele obținute se pot observa în Fig. 3.

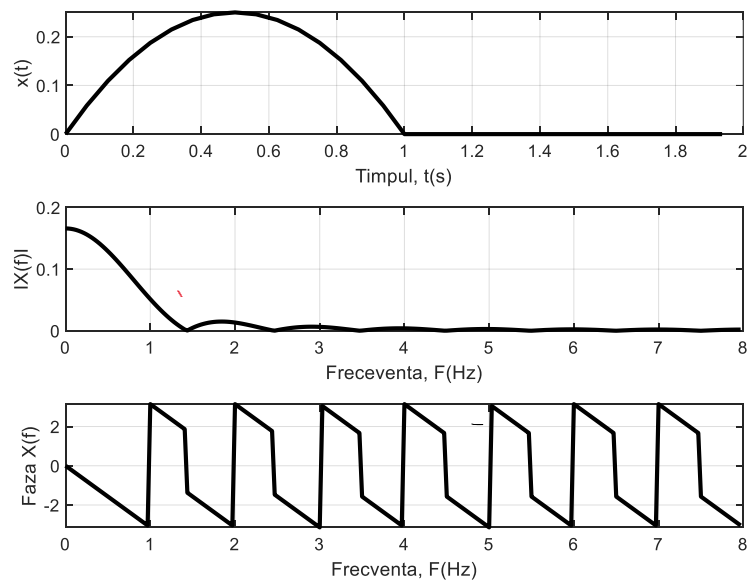


Fig. 2. Graficele Exemplului 1 folosind `fft()`

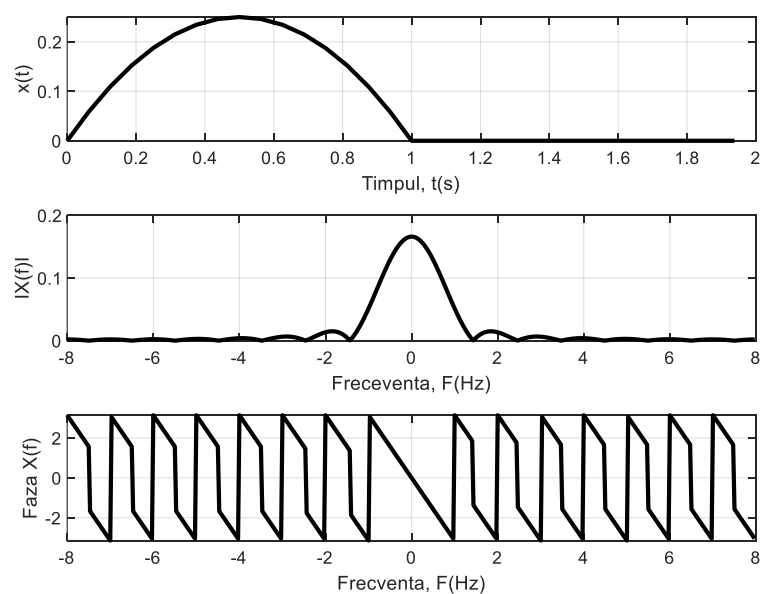


Fig. 3. Graficele Exemplului 1 folosind `fftshift()`

Exercițiu propus 1

Scrieți un program MATLAB care să afișeze modulul și faza Transformatei Fourier Discrete a semnalului:

$$x(t) = \begin{cases} 9 - (t - 3)^2, & 0 < t < 3 \\ 0, & \text{în rest} \end{cases}$$

1.3. Spectru semnalelor periodice folosind MATLAB

Așa cum a fost menționat în lucrarea 1, semnalele uzuale, întâlnite în practică, au un moment de apariție și un moment de dispariție, ceea ce înseamnă că semnale riguros periodice nu există în practică.

Totuși, în anumite situații, este util să se modeleze un semnal de durată finită, având pe durata sa de existență o variație de tip periodică, printr-o funcție periodică de timp. Această modelare nu conduce la erori dacă durata de existență a semnalului este mult mai mare decât perioada de repetiție și decât durata regimurilor tranzitorii apărute în circuit la aplicarea, respectiv suprimarea semnalului și, în plus, dacă nu interesează tocmai aceste regimuri tranzitorii.

Exemplu 2

Folosind Matlab reprezentați în timp semnalul de mai jos și spectrul de amplitudini al acestuia, în prezența și absența zgomotului:

$$x(t) = \begin{cases} 0.6 \cdot \sin(2\pi \cdot f_1 \cdot t) + \sin(2\pi \cdot f_2 \cdot t), & 0 < t < 50\text{ms} \\ 0, & \text{în rest} \end{cases}$$

```
clear
clc
close all
```

```
prompt = 'Caz=1 analiza semnal fara zgomot. Caz 2 semnal  
cu zgomot. Care este valoarea lui caz? :';
```

```
caz = input(prompt);
%%caz=1 semnal fara zgomot
%%caz=2 semnal cu zgomot
```

```
F1 = 50;
F2 = 120;
Fs = 1000; % frecventa de esantionare
```

```

Ts = 1/Fs; % perioada de esantionare
N = 1024; %lungimea transformatei Fourier
df = Fs/N; % rezolutia transformatei Fourier
Tmax = 0.05; %durata semnalului
t = 0: Ts :Tmax-Ts; % vector timp
F = 0:df:Fs/2-df; % vectorul de frecvente
%% semnal compus din doua semnale sinusoidale cu f1=50Hz
si f2= 120Hz.
x = 0.6*sin(2*pi*F1*t) + sin(2*pi*F2*t);
if caz == 1
plot(1000*t,x),grid
title('Semnalul neperturbat de zgomot')
xlabel('t (ms)')
ylabel('x(t)')

%% pause se foloseste daca se doreste o pauza in rulara
programului si se apară orice tasta pentru continuarea
rularii programului
% pause

%%FFT aplicata lui S
X = Ts*fft(x,N);
df = Fs/N;
% vectorul de frecvente:
figure (2)
plot(F,abs(X(1:N/2))),grid
title('Spectru de amplitudini pentru x(t)')
xlabel('F (Hz)')
ylabel('|X(F)|')
else
%% Perturbarea semnalului cu un semnal de zgomot.
xzg = x+0.5*randn(1,length(t));
plot(1000*t,xzg),grid
title('Semnalul perturbat de zgomot')
xlabel('t (ms)')
ylabel('x_zg(t)')

% pause

%%FFT aplicata lui S
Xzg = Ts*fft(xzg,N);
figure (2)

```

```

plot(F,abs(Xzg(1:N/2))),grid
title('Spectru de amplitudini pentru xzg(t)')
xlabel('F (Hz)')
ylabel('|X_zg(F)|')
end

```

După executarea codului prin selectarea cazului 1 se obțin graficele din Fig. 4, care conțin reprezentarea în timp a semnalului $x(t)$ neperturbat de zgomot și spectrul de amplitudini al acestuia calculat cu ajutorul funcției `fft()`.

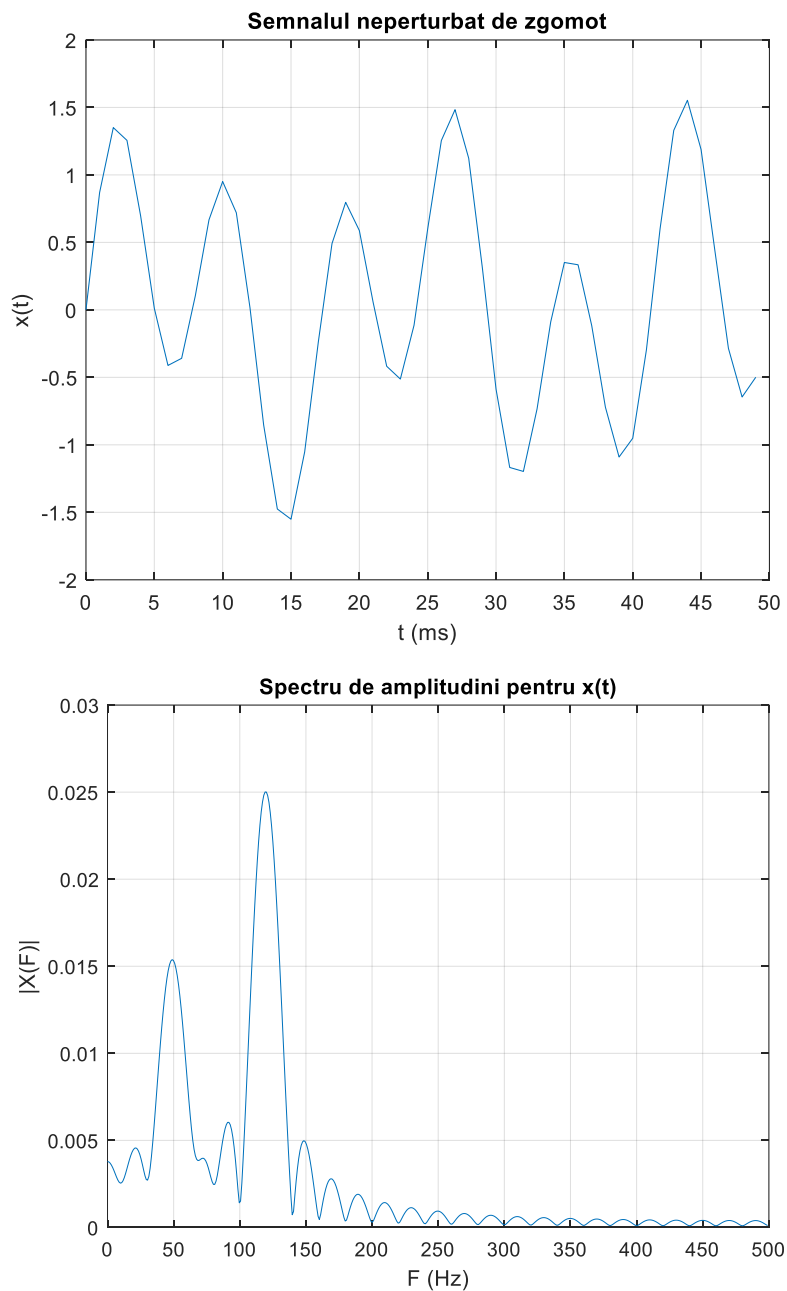


Fig. 4. Graficele Exemplului 2 folosind cazul 1 (semnal fără zgomot)

După executarea codului prin selectarea cazului 2 se obțin graficele din Fig. 5, care conțin reprezentarea în timp a semnalului $x(t)$ perturbat de zgomot și spectrul de amplitudini al acestuia calculat cu ajutorul funcției `fft()`.

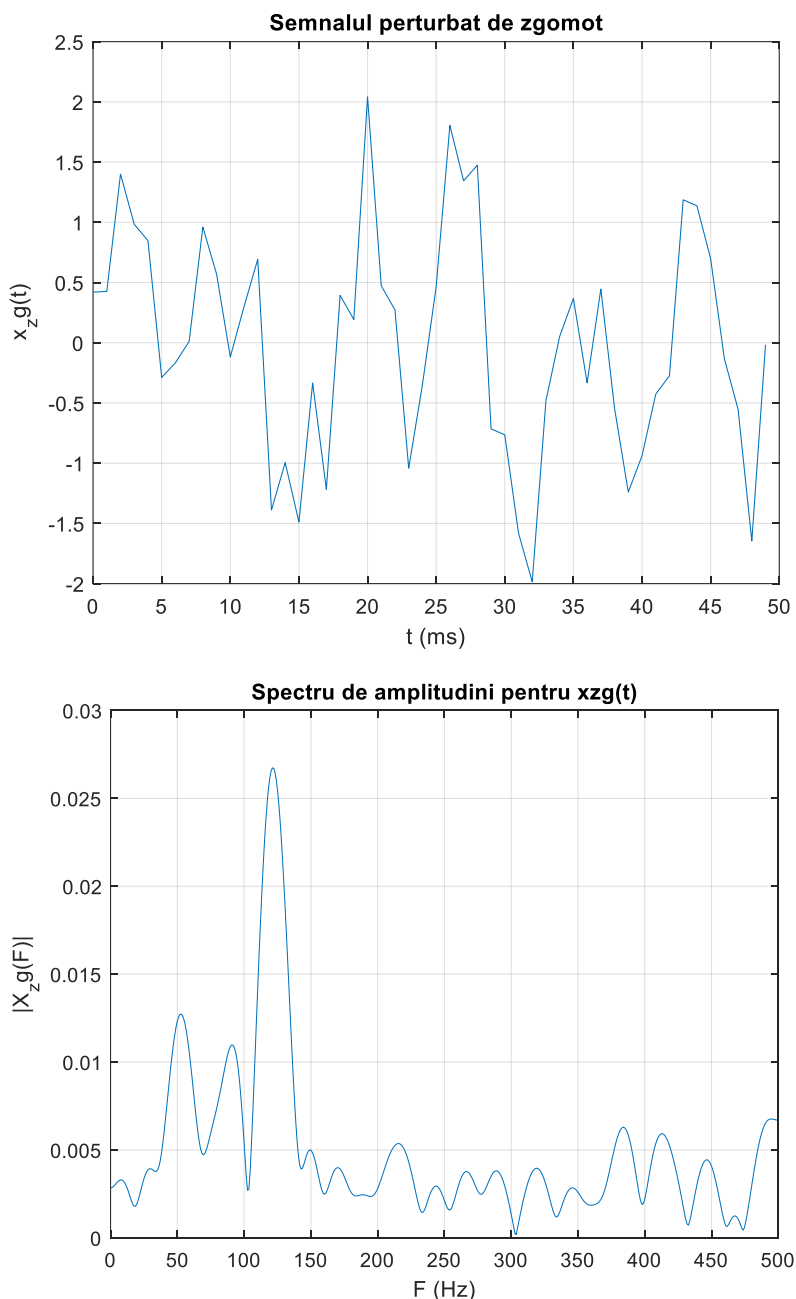


Fig. 5. Graficele Exemplului 2 folosind cazul 2 (semnal afectat de zgomot)

Exercițiu propus 2

Simulați programul de mai sus pentru frecvența de eșantionare setată la 500. Ce observați? Reveniți la $F_s=1000$ și modificați $T_{max}=0.5$. Ce observați? Dar dacă $F_s=500$ și $T_{max}=0.5$ ce se observă?

Exercițiu propus 3

Folosind MATLAB reprezentați în timp un semnal dreptunghiular periodic, cu factor de umplere 50% și spectrul de amplitudini al acestuia, în prezența și absența zgomotului (pentru reprezentarea în timp folosiți funcțiile din lucrarea anterioară). Schimbați factorul de umplere la 25%, respectiv 10%.

Exercițiu propus 4

Folosind MATLAB reprezentați în timp un semnalul triunghiular și spectrul de amplitudini al acestuia, în prezența și absența zgomotului (pentru reprezentarea în timp folosiți funcțiile din lucrarea anterioară).

Bibliografie

1. Mateescu, Adelaida, Dumitriu, N., Stanciu, L., **Semnale, circuite și sisteme**, Teora, București, 2001.
2. Introducere în prelucrarea semnalelor folosind Python / Valentin-Adrian Niță, Radu Alexandru Badea, Răzvan-Eusebiu Crăciunescu. – Timișoara : Editura Politehnica, 2022