Piotr Chojnacki:

MATLAB

Program na zaliczenie: „Odejmowanie widm”

{Poniższy program ma za zadanie odjęcie dwóch widm od siebie. Do poprawnego działania programu potrzebne są trzy funkcje: odejmowaniewidm.m współczynnik.m numer.m. Poniżej podane są kody źródłowe tych funkcji. W folderze skompresowanym znajdują się 4 przykładowe widma na których można przećwiczyć program. Również poniżej przedstawiam wyniki działania.}

{została zachowana oryginalna kolorystyka składni MATLABa}

```
function [k]=odejmowaniewidm(x,y);

% Piotr Chojnacki
% III rok informatyka chemiczna
% Wrocław dn. 22 maja 2005

% zadaniem funkcji jest odjęcie dwóch widm od siebie
% funkcję wywołuje się w sposób następujący: odejmowaniewidm(x,y)
% gdzie 'x'- oznacza widmo pierwsze 'y' - widmo drugie
% widma wczytuje się poprzez podanie samej nazwy bez rozszerzenia
% natomiast k jest parametrem zwracanym przez funkcję i jest to obliczony
% współczynnik o który odjęte są widma z równania: w1-k*w2

% przykłady wywołań    odejmowaniewidm('OCT2RS20','OCT2RS85');
%                      odejmowaniewidm('NBUT20','NBUT80');
% przykład wywołania funkcji dla widm stężeniowych
%                      odejmowaniewidm('STEZ01','STEZ02');

global z intensx intensy n m widmox widmoy
% deklaracja zmiennych globalnych wykorzystywanych w drugiej funkcji
disp([' ']);
```

```
disp(['---------------------------------------- ']);
disp('Funkcja odejmie teraz dwa widma od siebie.']);
disp(['---------------------------------------- ']);
disp([' ']);
hold on % włączenie rysowania wszystkich widm na jednym wykresie

eval(['load ' x '.PRN']); % wczytanie pierwszego widma
dane=eval(x);             % i dodanie rozszerzenia .PRN
liczfalx=dane(:,1);
intensx=dane(:,2);
widmox=[dane(:,1) dane(:,2)];


plot(liczfalx,intensx,'b');    % rysowanie I widma w kolorze niebieskim
title('widma');                % tytuł wykresu
xlabel('czestosc (liczby falowe)');      % nazwa osi x
ylabel('absorbancja (intensywnosc)'); % nazwa osi y
grid on    % włączenie siatki na wykresie


eval(['load ' y '.PRN']); % wczytanie drugiego widma
dane1=eval(y);            % i dodanie rozszerzenia .PRN
liczfaly=dane1(:,1);
intensy=dane1(:,2);
widmoy=[dane1(:,1) dane1(:,2)];

plot(liczfaly,intensy,'g');        % rysowanie II widma w kolorze zielonym
legend('widmo I','widmo II'); % legenda
disp(['Zostało narysowane pierwsze i drugie widmo ']);
disp([' ']);

% ------------------------------------------------------------------------
% OFFSET
% wynonywanie offsetu czyli odejmowanie od widma wybranej
% intensywności dla danej liczby falowej
% po odjęciu wartości intensywności przy danej liczbie falowej otrzymujemy
% dokładniejszą wartośc współczynnika k
% ------------------------------------------------------------------------

lf=input('Podaj liczbę falową do przeprowadzenia offsetu: ');

po=numer(widmoy(:,1),lf); % odjęcie wartości intensywności przy danej
                          % liczbiefalowej od  widma pierwszego
```

```
intensy=intensy-intensy(po);
po=numer(widmox(:,1),lf); % odjęcie wartości intensywności przy danej liczbie
                          % falowej od widma drugiego
intensx=intensx-intensx(po);

% ------------------------------------------------------------------------
% od tego momentu wartości intensx i intensy sa podstawione wartosciami po
% odjeciu intensywności przy danej liczbie falowej
% ------------------------------------------------------------------------

figure       % nowe widma po offset rysowane są w nowym oknie
hold on      % dopisywanie do tego samego okna nowych wykresów
grid on      % włączenie siatki na wykresie


plot(liczfalx,intensx,'b');    % rysowanie I widma w kolorze niebieskim
title('widma po offset');                % tytuł wykresu
xlabel('czestosc (liczby falowe)');   % nazwa osi x
ylabel('absorbancja (intensywnosc)'); % nazwa osi y


plot(liczfaly,intensy,'g');   % rysowanie II widma w kolorze zielonym
legend('widmo I  - po offset','widmo II - po offset'); % legenda
disp(['Zostało narysowane pierwsze i drugie widmo po dokonaniu offsetu ']);


% przedział liczenia współczynnika k
disp([' ']);
disp(['Wybierz przedział dla którego zostanie policzony współczynnik']);
disp(['o który zostaną odjęte widma według równania w1-k*w2 ']);
disp([' ']);
% zerowanie zmiennych oznaczających granicę
% przedziału szukania współczynnika k
n=0;
m=0;

n=input('poczatek przedziału:  '); % wczytanie początku przedziału
m=input('koniec przedziału:  ');    % wczytanie końca przedziału

k=1; % początkowa wartość wspołczynnika k

wspol=fminsearch('wspolczynnik',k); % wywołanie funkcji liczącej
                                    wspołczynnik k
```

```
k=abs(wspol) % zapisywanie wyniku funkcji wspolczynnik pod zmienną k w
             % module tak żeby był zawsze liczbą dodatnią
odjete= [widmox(:,1) (intensx-k*intensy)]; % macierz złożona z dwóch
                                             kolumn: liczby falowe i
                                             intensywność po odjęciu
plot(widmox(:,1),odjete(:,2),'r');  % rysowanie odjętego widma w kolorze
                                      czerwonym

legend('widmo I  - po offset','widmo II - po offset','widmo po odjeciu');
% legenda na wykresie

disp(['Na czerwono zostało narysowane widmo po odjęciu']);
disp([' ']);
%save odejmowaniewidm % zapisanie zmiennych z okna Workspace
                      % pod nazwą odejmowaniewidm.mat
```

```
function f=wspolczynnik(k);

% Piotr Chojnacki
% III rok informatyka chemiczna
% Wrocław dn. 22 maja 2005

% funkcja oblicza współczynnik k korzystając z funkcji numer.m podającej
% numer elementu o danej liczbie falowej

global z intensx intensy n m widmox widmoy
% deklaracja uzycia globalnych zmiennych z funkcji odejmowaniewidm.m

pp=0; % zerowanie zmienych oznaczających początek i koniec
kk=0;

pp=numer(widmox(:,1),n); % podanie numeru elementu za pomocą funkcji
                         % numer
kk=numer(widmoy(:,1),m);

% obliczanie sumy kwadratu odchyleń
% w funkcji odejmowaniewidm.m nadana była początkowa wartość
% współczynnika k
% k=1
```

```
% w tym momencie funkcja liczy sumę i sama dobiera współczynnik k
% wynikiem jest współczynnik k, dla którego otrzymujemy najmniejszą wartość
% sumy
suma=0;
for i=kk:pp
   suma1=((intensx(i,1))-k*(intensy(i,1))).^2;
   suma=suma+suma1;
end
f=suma; % podstawienie pod funkcję obliczonej sumy
```

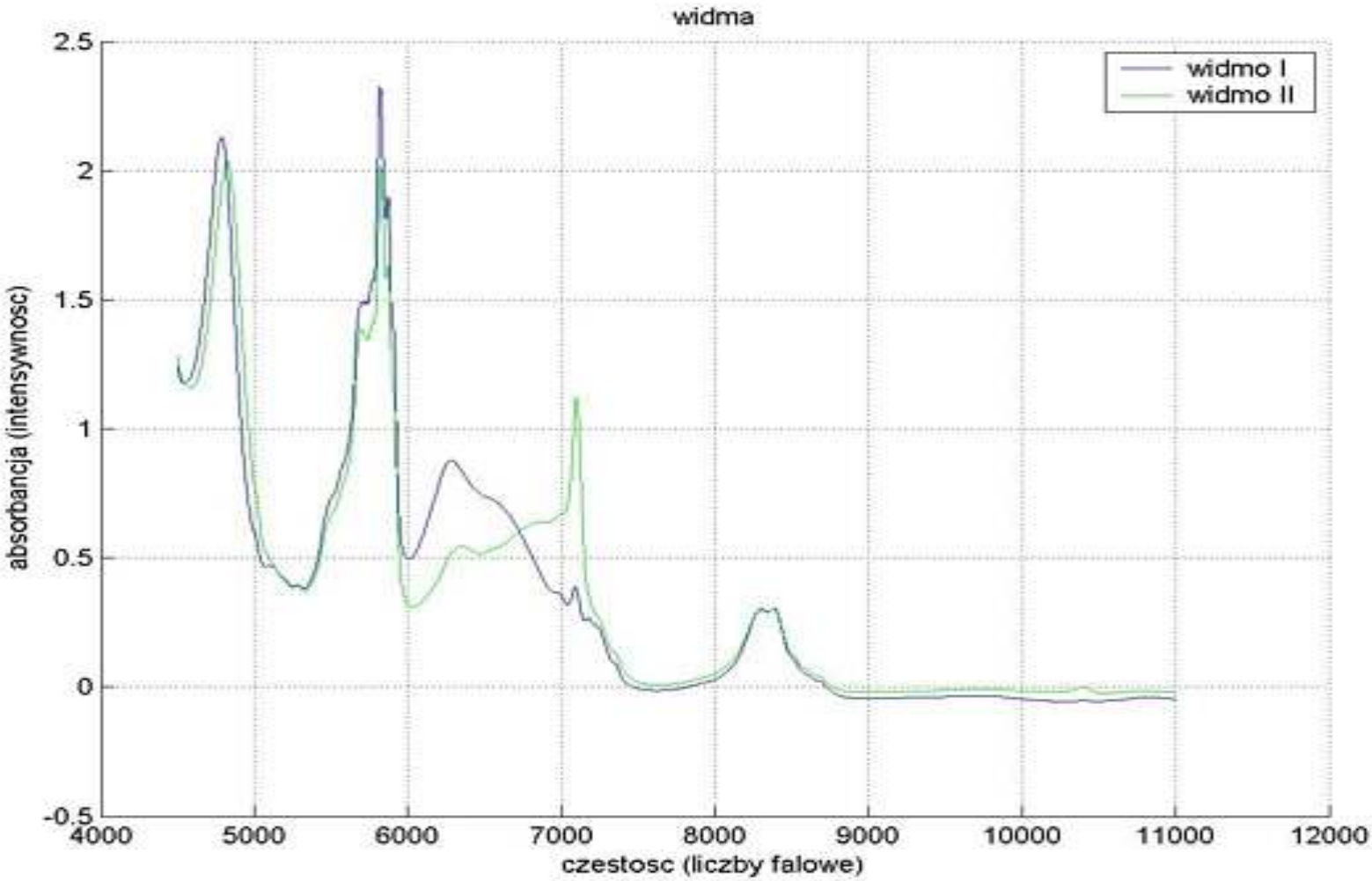
```
function pos=numer(x,lf)

zp=length(lf);
if zp==1
   xp=abs(x-lf);
   [pom pos]=min(xp);
else
   for i=1:zp
      xp=abs(x-lf(i));
      [pom pos(i)]=min(xp);
   end
end
```

{Przykładowe wywołanie programu dla widm NBUT20 i NBUT80}

>> odejmijwidma('NBUT20','NBUT80');

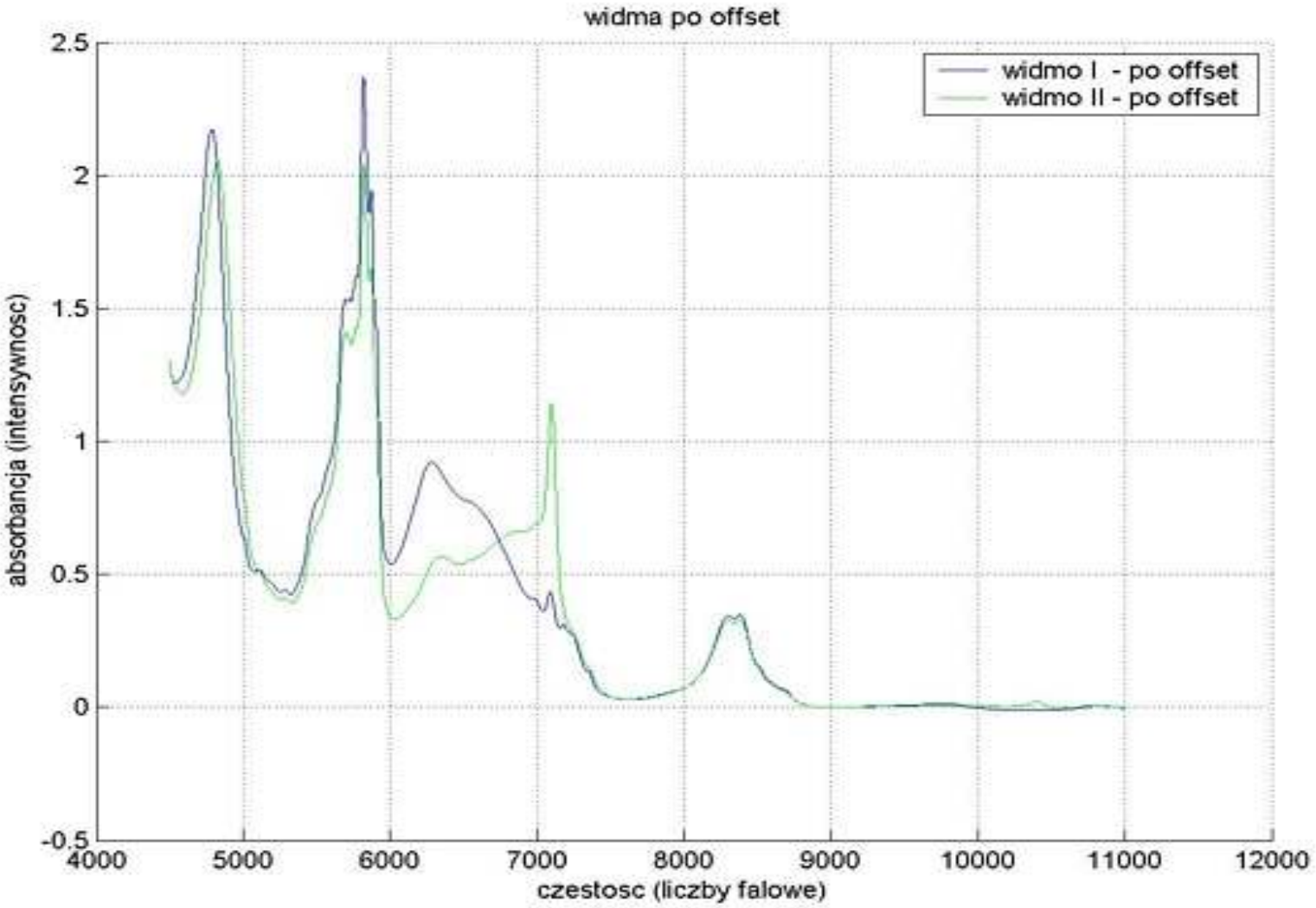
Funkcja odejmie teraz dwa widma od siebie.

Zostało narysowane pierwsze i drugie widmo



Podaj liczbę falową do przeprowadzenia offsetu: 9000 {w przypadku tego widma jest to najlepsza wartość}

Zostało narysowane pierwsze i drugie widmo po dokonaniu offsetu



Wybierz przedział dla którego zostanie policzony współczynnik
o który zostaną odjęte widma według równania w1-k*w2

poczatek przedziału: 8000 {najlepszy przedział to 8000-9000. Dobrze jest jak widma się w takim przedziale nakładają wtedy współczynnik jest bliski 1 i widma są w całości odjęte.}

koniec przedziału: 9000

k =

1.0499

Na czerwono zostało narysowane widmo po odjęciu