Hleb Shypula  
286720  
 17 czerwca 2020

**Sprawozdanie**

Pakiet obliczeniowy MATLAB i jego zastosowania

1. **Laboratorium 1 Podstawy**

**prog.m**

% tworze zapis

prompt = 'Podaj wektor [x(1), . . . , x(i)]: ';

% wyswietlam zapis i pobieram wektor

v = input(prompt);

% tworze zapis

prompt = 'Podaj funkcje - 1 (sin) lub 2 (cos) ';

% wyswietlam zapis i pobieram przel

przel = input(prompt);

% wywoluje funkcje, ktora pryjmuje wektor i przel i zwraca [...]

[maks, minim, parz, niep, niezer, moj\_fun] = przetworz(v, przel);

% jezeli przel == 1 lub 2, to wyswietlam wyniki korzystajac z disp()

if przel==1 || przel==2

disp(' ');

disp('Wartosc maksymalna elementow wektora: ');

disp(maks);

disp('Wartosc minimalna elementow wektora: ');

disp(minim);

disp('Liczba elementow parzystych w wektorze: ');

disp(parz);

disp('Liczba elementow nieparzystych w wektorze: ');

disp(niep);

disp('Liczba elementow niezerowych w wektorze: ');

disp(niezer);

if przel==1

disp('Wektor sin(v): ');

disp(moj\_fun);

elseif przel==2

disp('Wektor cos(v): ');

disp(moj\_fun);

end

end

**przetworz.m**

% potrzebna funkcja

function [maks, minim, parz, niep, niezer, moj\_fun] = przetworz(v, przel)

% jezeli przel ~= 1 i 2, to wyswietlam komunikat o bledzie

% nadaje wartosci nan naszym elementom na wyjsciu na wszelki wypadek

% i return'em koncze dzialanie programu

if przel ~= 1 && przel ~= 2

disp('ERROR - przel ~= 1 lub 2 ');

maks = nan; minim = nan; parz = nan;

niep = nan; niezer = nan; moj\_fun = nan;

return;

end

% szukam maksa

maks = max(v);

% szukam mina

minim = min(v);

% licze ilosc parzystych elementow

parz = 0;

s = size(v);

for i=1:s(2)

if mod(v(i),2)==0

parz=parz+1;

end

end

% licze ilosc nieparzystych elementow

niep = 0;

s = size(v);

for i=1:s(2)

if mod(v(i),2)==1

niep=niep+1;

end

end

% licze niezerowe elementy

v\_niezer = find(v);

s\_niezer = size(v\_niezer);

niezer = s\_niezer(2);

% zaleznie od przel, do moj\_fun zapisuje odpowiedni wektor

if przel==1

moj\_fun = sin(v);

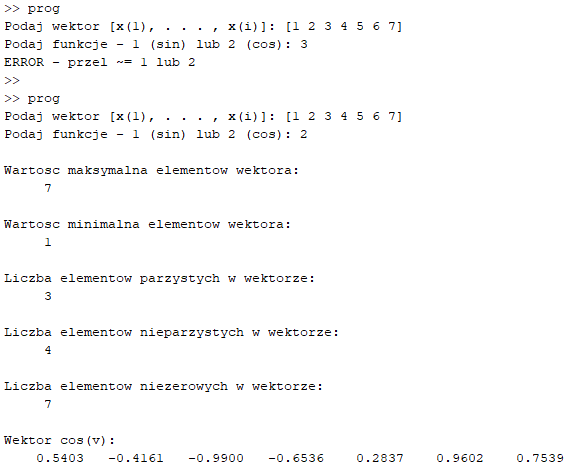
elseif przel==2

moj\_fun = cos(v);

end

end

**przykładowe wyniki**



1. **Laboratorium 2 Wykresy**

**Zad 1a**

% zakres zmiennej t

t = linspace(0,2\*pi);

% funkcja sin

y\_sin = sin(t);

% funkcja cos

y\_cos = cos(t);

% rysuje w jednym wywolaniu funkcji plot

plot(t,y\_sin,'-k',t,y\_cos,'--b')

% siatka

grid on;

%podpis OX

xlabel('t');

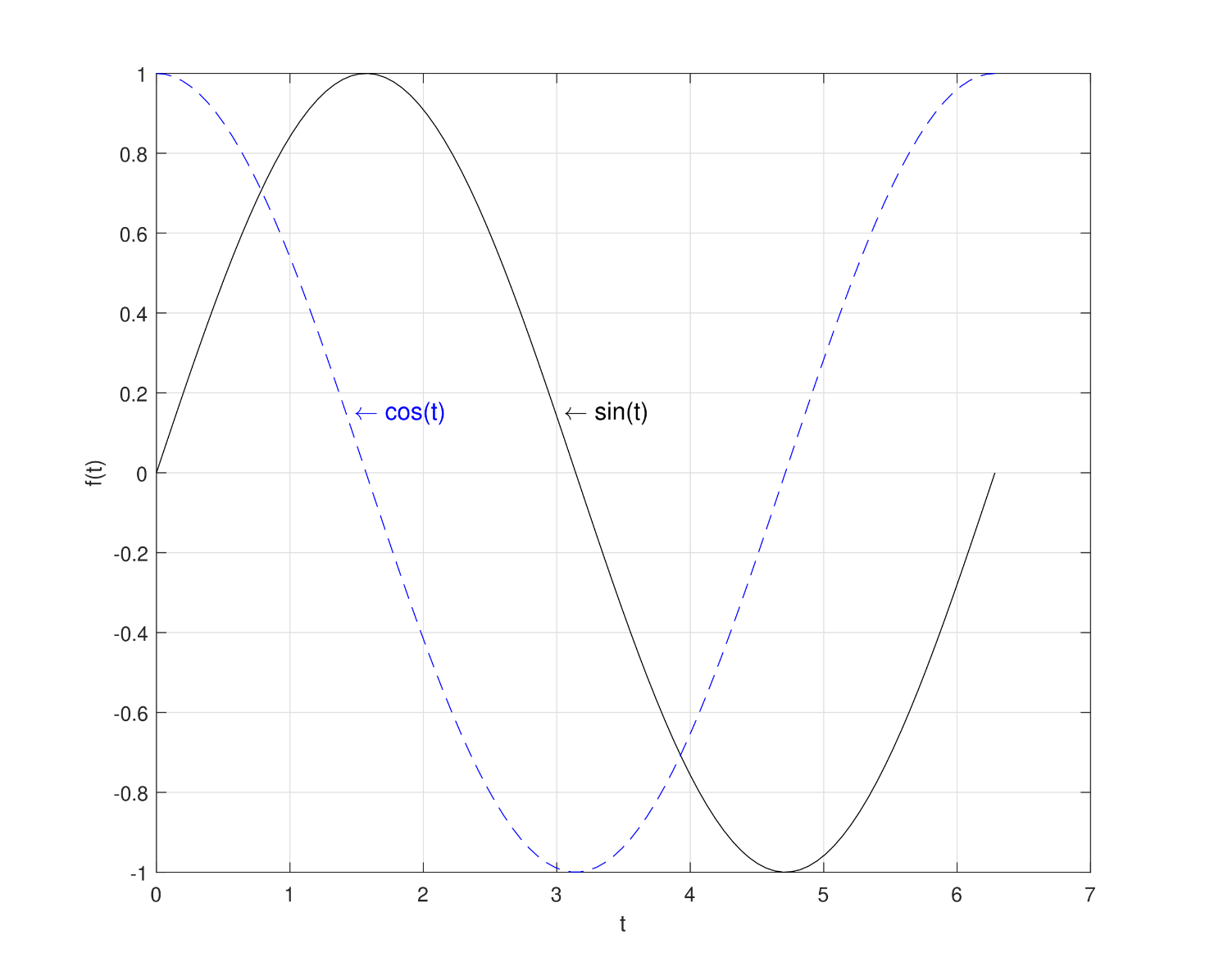
% podpis OY

ylabel('f(t)');

% podpis krzywych

text(3.05,0.16,'\leftarrow sin(t)','Color','black','FontSize',12);

text(1.48,0.16,'\leftarrow cos(t)','Color','blue','FontSize',12);



**Zad 1b**

% zakres zmiennej t

t = linspace(0,2\*pi);

% funkcja sin

y\_sin = sin(t);

% funkcja cos

y\_cos = cos(t);

% rysuje w dwoch wywolaniach funkcji plot

plot(t,y\_sin,'-k')

hold on

plot(t,y\_cos,'--b')

% siatka

grid on;

%podpis OX

xlabel('t');

% podpis OY

ylabel('f(t)');

% podpis krzywych

text(3.05,0.16,'\leftarrow sin(t)','Color','black','FontSize',12);

text(1.48,0.16,'\leftarrow cos(t)','Color','blue','FontSize',12);

**Zad 2**

% rysuje dzielac przedzial na 4 podprzedzialy

subplot(2,2,1)

% zakres zmiennej t

t = linspace(-2\*pi,2\*pi);

y\_sin = sin(t);

plot(t,y\_sin)

% siatka

grid on;

%podpis OX

xlabel('t');

% podpis OY

ylabel('f(t)');

% podpis wykresu

title('sin(t)')

subplot(2,2,2)

t = linspace(-2\*pi,2\*pi);

y\_cos = cos(t);

plot(t,y\_cos)

grid on;

%podpis OX

xlabel('t');

% podpis OY

ylabel('f(t)');

title('cos(t)')

subplot(2,2,3)

t = linspace(-pi/2,pi/2);

y\_tan = tan(t);

plot(t,y\_tan)

grid on;

% zakres po OY

ylim([-10 10]);

%podpis OX

xlabel('t');

% podpis OY

ylabel('f(t)');

title('tan(t)')

subplot(2,2,4)

t = linspace(0,pi);

y\_ctg = cot(t);

plot(t,y\_ctg)

grid on;

ylim([-10 10]);

% zakres po OX

xlim([-1 4]);

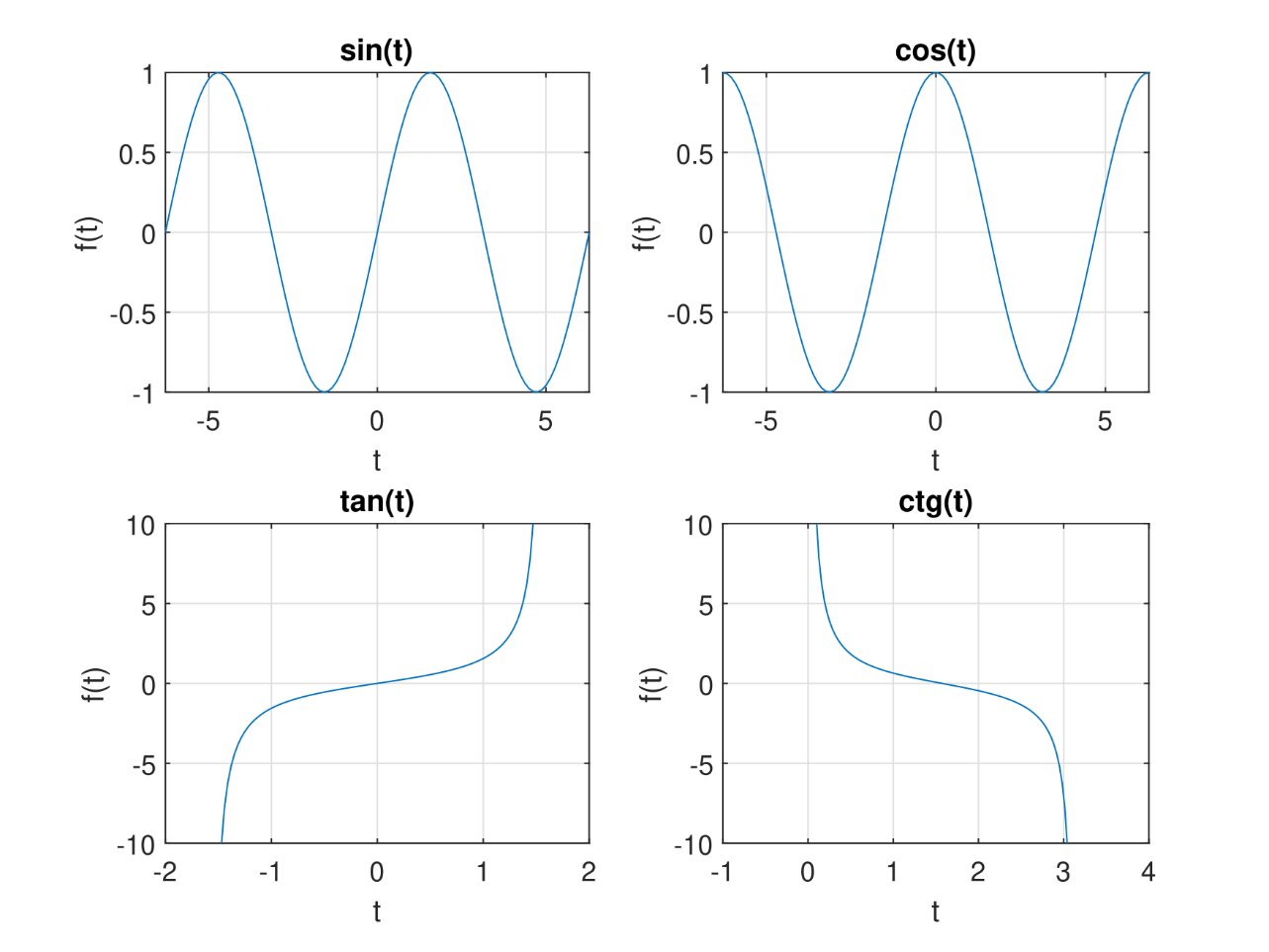
%podpis OX

xlabel('t');

% podpis OY

ylabel('f(t)');

title('ctg(t)')



**Zad 3**

1 'Luksemburg' 106705  
2 'Irlandia' 78785  
3 'Norwegia' 74356  
4 'Szwajcaria' 64649  
5 'Holandia' 56383  
6 'Islandia' 55917  
7 'Szwecja' 52984  
8 'Niemcy' 52559  
9 'Austria' 52137  
10 'Dania' 52121

% odczytuje dane

pkb = readtable('PKB.dat');

% rysuje wykres

bar(pkb.Var1, pkb.Var3,'FaceColor',[0.863 0.078 0.235],...

'EdgeColor',[178/255 34/255 34/255],'LineWidth',2);

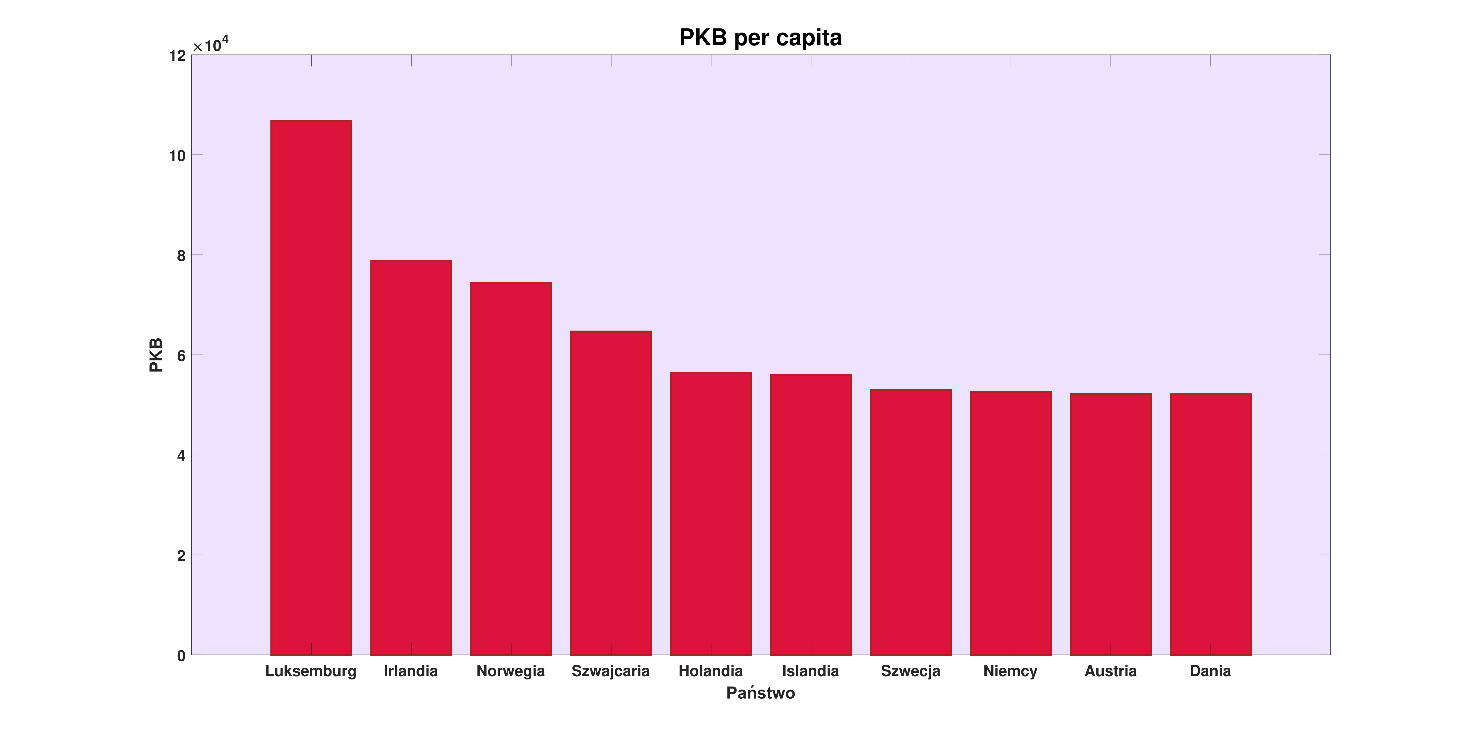
xlabel('Państwo');

ylabel('PKB');

title('PKB per capita');

% zmieniam indeksy na kraje

set(gca, 'XTick', 1:10, 'XTickLabel', pkb.Var2);



**Zad 4**

% pobieram dane

pol = readtable('Poland.dat');

gre = readtable('Greece.dat');

% tworze animacje

h = figure;

filename = 'zad4.gif';

p = animatedline('Color','b','LineWidth',3);

g = animatedline('Color','r','LineWidth',3);

% ustaliam zeby miec statyczne pole

set(gca,'XLim',[2000 2018],'YLim',[0 35000]);

% siatka

grid on;

% podpisy

xlabel('Rok');

ylabel('PKB, $');

title('PKB per capita animation');

% legenda

legend({'Polska','Grecja'},'Location','northeast','Orientation','horizontal')

% petla do rysowania ramek

for i = 1:length(pol.Var1)

addpoints(p,pol.Var1(i),pol.Var2(i));

addpoints(g,gre.Var1(i),gre.Var2(i));

drawnow

frame = getframe(h);

im = frame2im(frame);

[imind,cm] = rgb2ind(im,256);

% zapis do gif

if i == 1

imwrite(imind,cm,filename,'gif', 'Loopcount',inf,'DelayTime',0.2);

else

imwrite(imind,cm,filename,'gif','WriteMode','append','DelayTime',0.2);

end

end

[Obrazek .gif z animacją (kliknij)](https://s7.gifyu.com/images/zad4.gif)

**Zad 5**

% przedzialy

[X,Y] = meshgrid(-3:0.1:3,-4:0.1:4);

% funkcja

Z = sin(X) .\* cos(Y);

% wykres

surf(X,Y,Z)

% podpis OX

xlabel('x');

% podpis OY

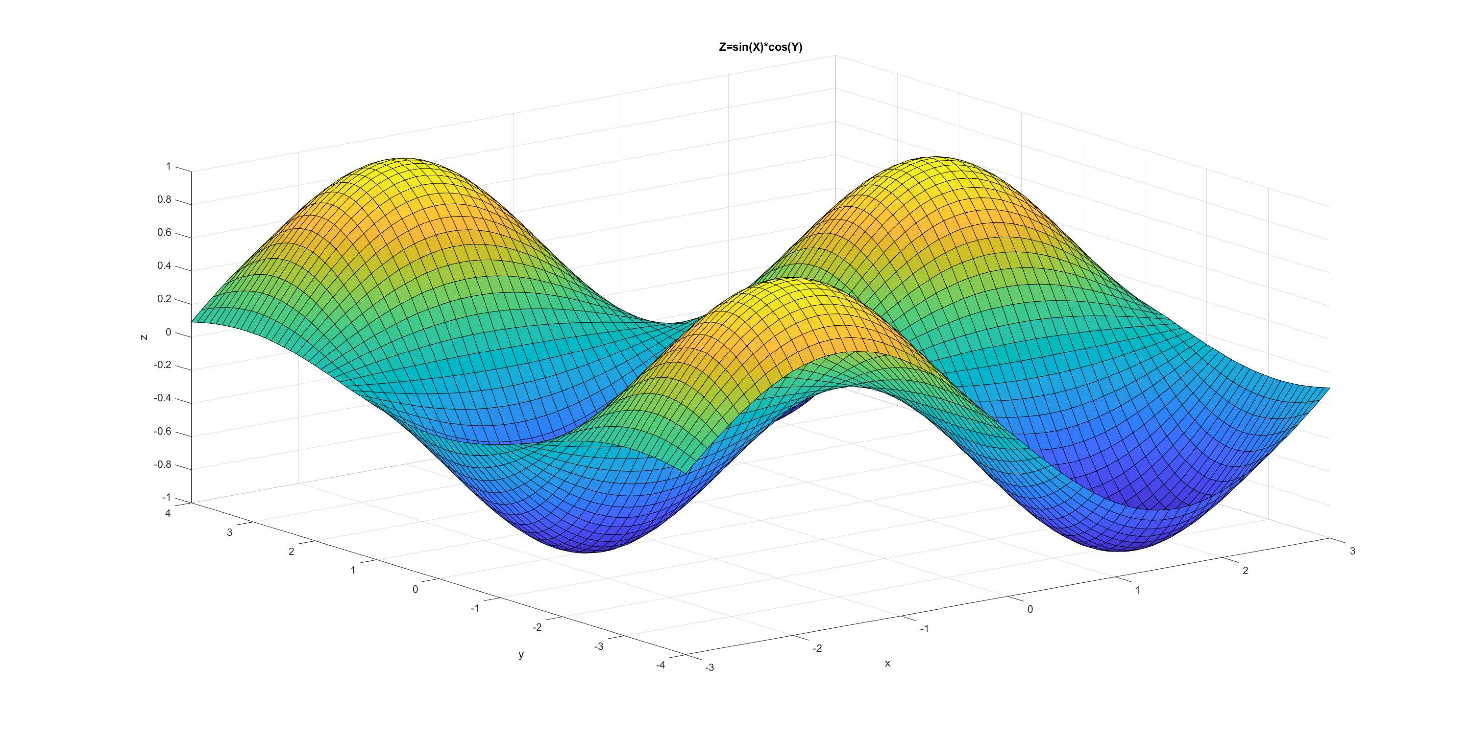
ylabel('y');

% podpis OZ

zlabel('z');

% tytul

title('Z=sin(X)\*cos(Y)');



1. **Laboratorium 3 GUI**

**gui.m**

function varargout = gui(varargin)

% GUI MATLAB code for gui.fig

% GUI, by itself, creates a new GUI or raises the existing

% singleton\*.

%

% H = GUI returns the handle to a new GUI or the handle to

% the existing singleton\*.

%

% GUI('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in GUI.M with the given input arguments.

%

% GUI('Property','Value',...) creates a new GUI or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before gui\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to gui\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui

% Last Modified by GUIDE v2.5 23-May-2020 14:53:31

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @gui\_OpeningFcn, ...

'gui\_OutputFcn', @gui\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before gui is made visible.

function gui\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to gui (see VARARGIN)

% Choose default command line output for gui

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes gui wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = gui\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% wychodze po kliknieciu

close;

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% pobieram min, max i krok

min = str2double(get(handles.edit1, "String"))

max = str2double(get(handles.edit2, "String"))

krok = str2double(get(handles.edit3, "String"))

% tworze x

x = min:krok:max

% pobieram wybor

contents = cellstr(get(handles.popupmenu1, "String"))

pop\_choice = contents{get(handles.popupmenu1, "Value")}

% rysuje w zaleznosci od wyboru

if(strcmp(pop\_choice, 'sin(x)'))

plot(x,sin(x));

ylabel("sin(x)");

elseif(strcmp(pop\_choice, 'cos(x)'))

plot(x,cos(x));

ylabel("cos(x)");

elseif(strcmp(pop\_choice, 'tan(x)'))

plot(x,tan(x));

ylabel("tg(x)");

elseif(strcmp(pop\_choice, 'ctg(x)'))

plot(x,cot(x));

ylabel("ctg(x)");

end

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.

function popupmenu1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit2\_Callback(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text

% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.

function edit2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit3\_Callback(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text

% str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.

function edit3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

1. **Laboratorium 4 Obliczenia numeryczne**

% zad 1

A = readmatrix('L4\_mac\_A.txt');

B = readmatrix('L4\_mac\_B.txt');

C=A\*B;

xlswrite ('L4\_mac\_C.xls', C, 'wynik');

% zad 2

num=quad('sin', 0, pi/2);

syms x;

sym=int(sin(x), 0, pi/2);

sym1=int(sin(x));

% zad 3

syms x;

f=4\*x^7+5\*x^4+cos(2\*x);

wynik=diff(f);

% zad 4

syms x y

eqns = [2\*x+2\*y==-6, 10\*x-5\*y==30];

vars = [y x];

[soly, solx] = solve(eqns,vars);

% zad 5

syms y(x);

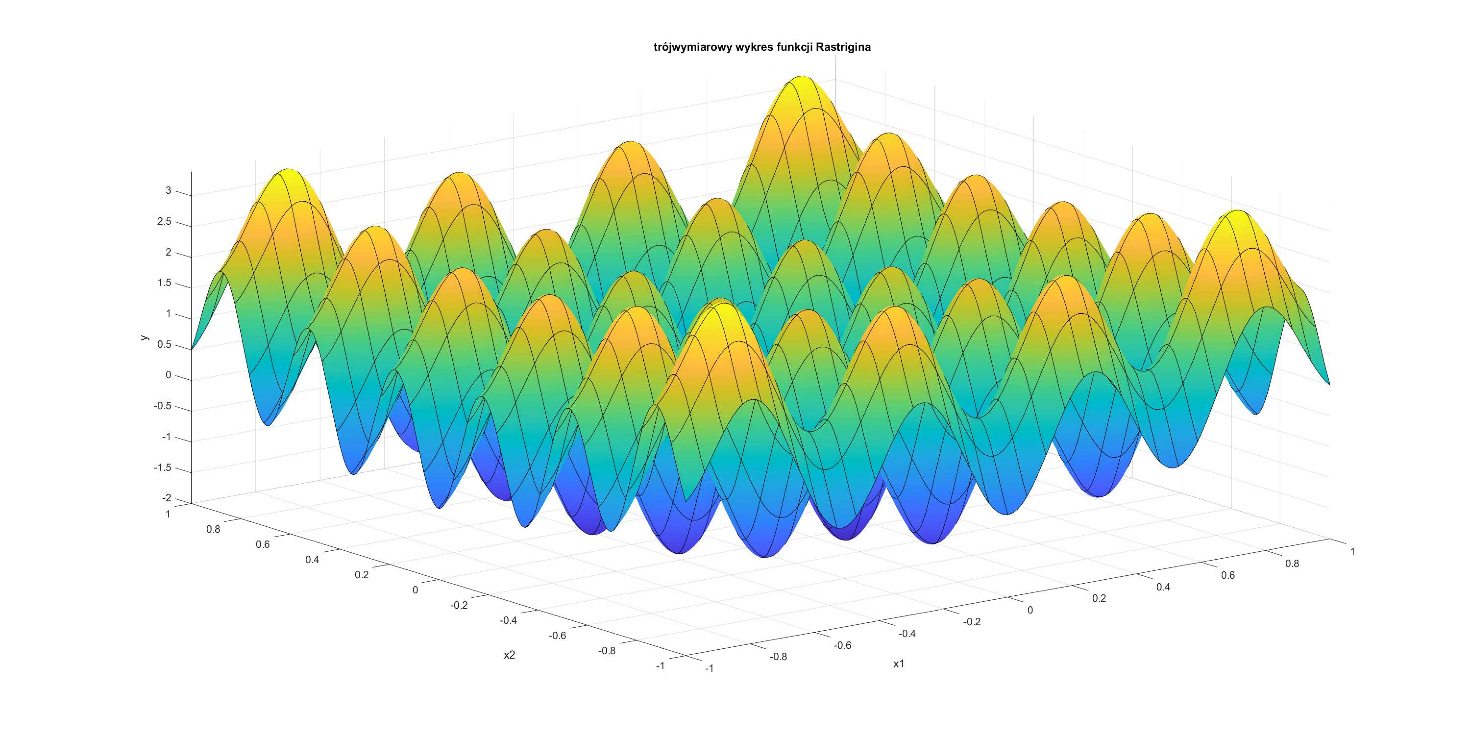
dsolve(diff(y)==-2\*x\*y);

% zad 6

syms x1 x2;

y=x1^2+x2^2-cos(12\*x1)-cos(18\*x2);

fsurf(y, [-1,1]);

****

1. **Laboratorium 5 Zastosowania**

clear;

set(gcf, 'Position', get(0, 'Screensize'));

[t,y] = ode45(@row\_1, [0 1], [1 0.5 3]);

plot(t,y,'-\*');

function dy = row\_1(t,y)

dy = zeros(3,1);

dy(1) = -y(1) + 3\*y(3);

dy(2) = -y(2) + 2\*y(3);

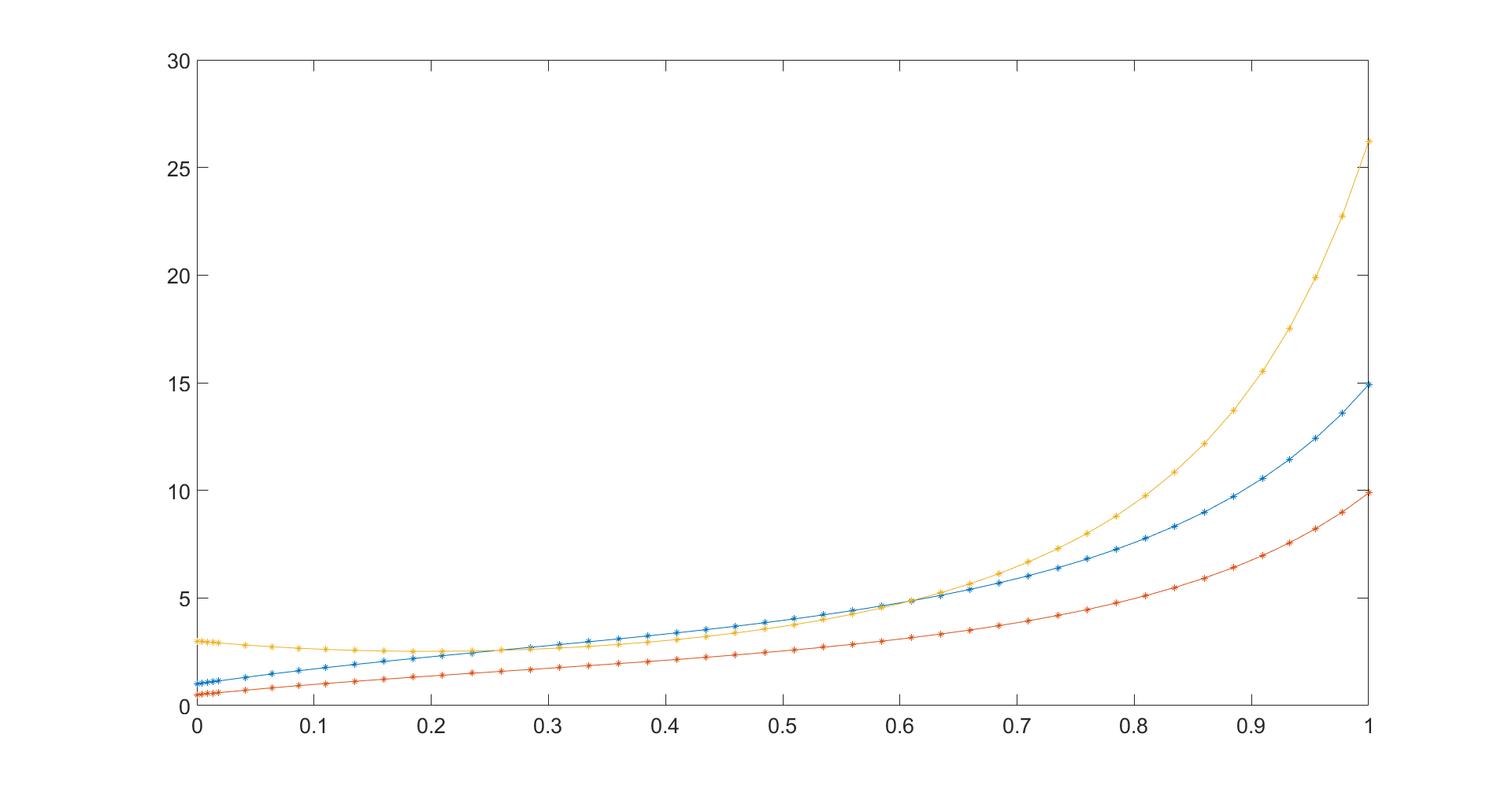
dy(3) = y(1)\*y(1) - 2\*y(3);

end

function f = rastrigin(x)

f = x(1)^2 + x(2)^2 - cos(12\*x(1)) - cos(18\*x(2))

end



**Projekt**

1. **Komentarze**

Jako dane do wykresów stosowałem przeważnie dane dotyczące Europy oraz Polski, ponieważ aktualnie znajdujemy się tutaj i otrzymane wyniki będą bardziej interesujące.

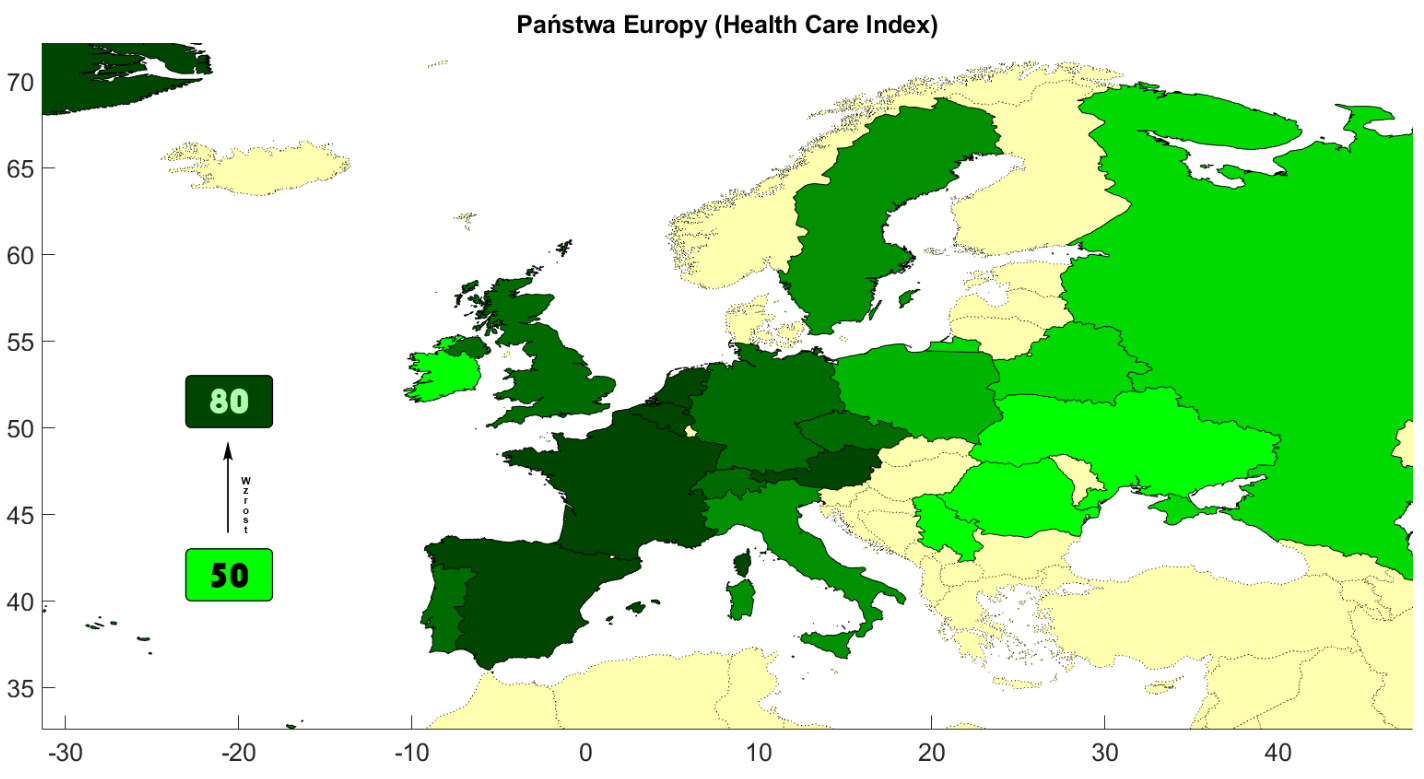
Myśle, że Panu nie chodziło o dokładność i konkretność danych, dlatego podchodziłem twórczo.

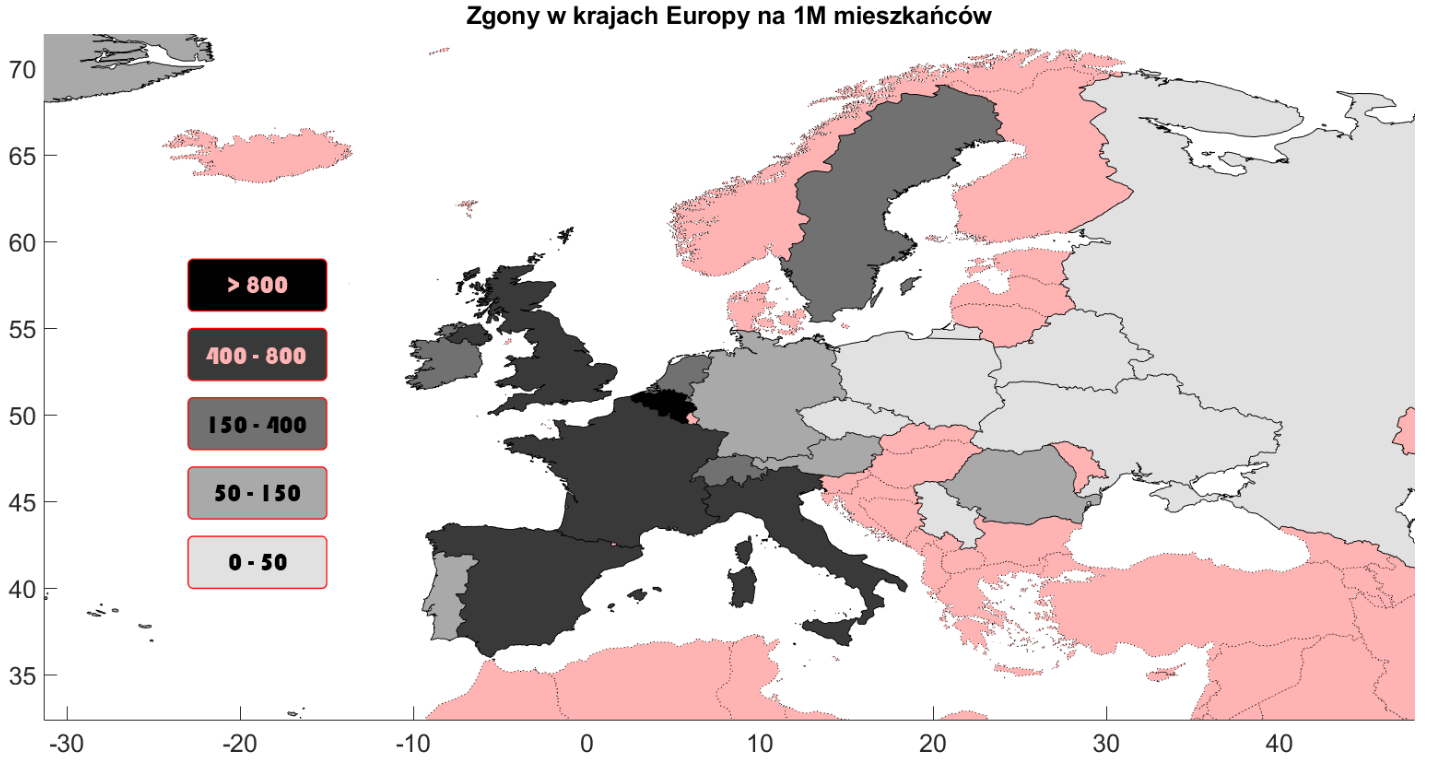
Wszystkie dane pobierałem w dniach 25-30 maja.

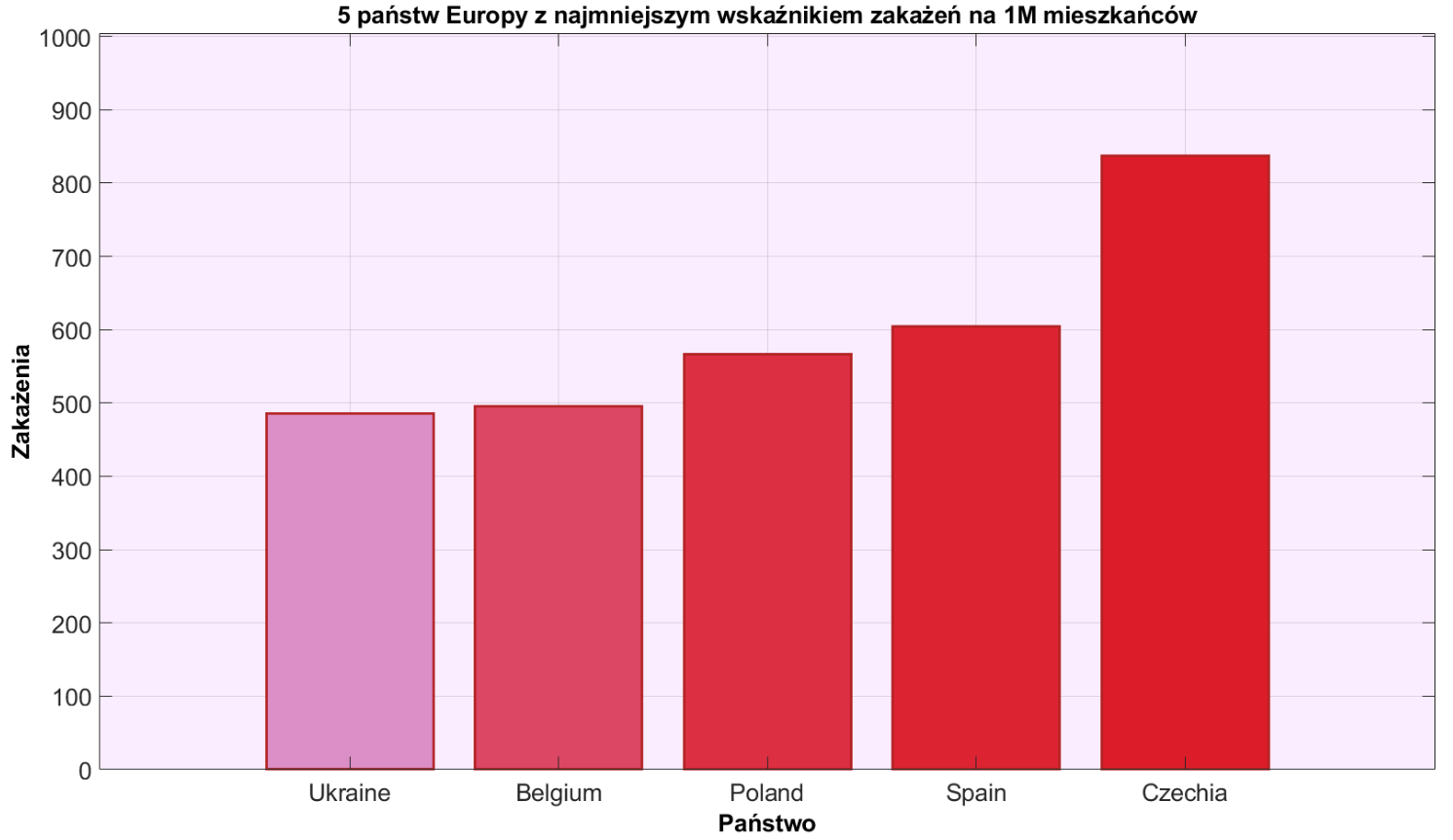
Stosowałem przeliczanie danych na 1 milion mieszkańców, myślę, że taka skala będzie bardziej przyjemna.

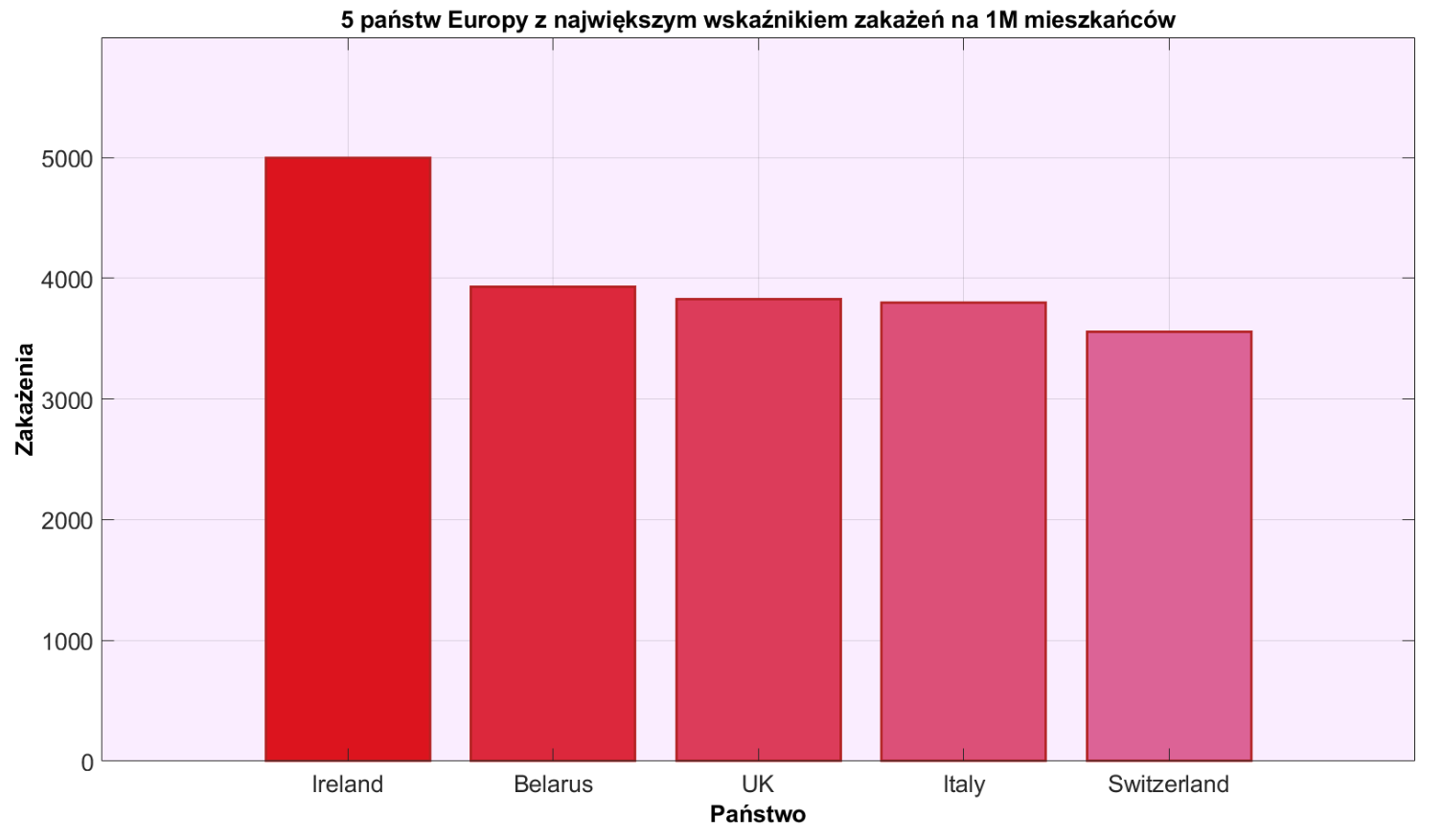
Plik źródłowy można pobrać [tutaj](https://bit.ly/2zTO5jk).

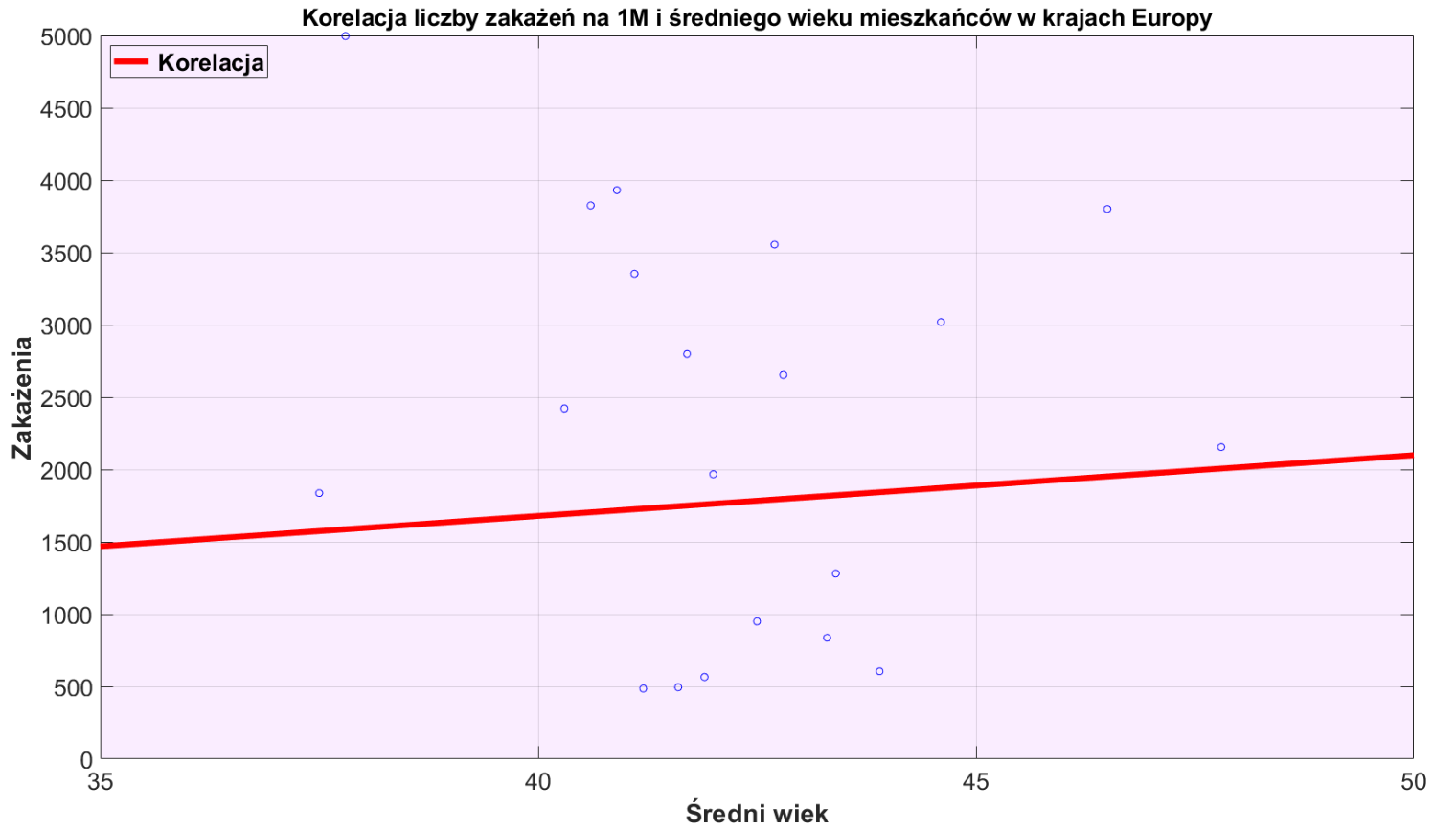
1. **Wyniki**

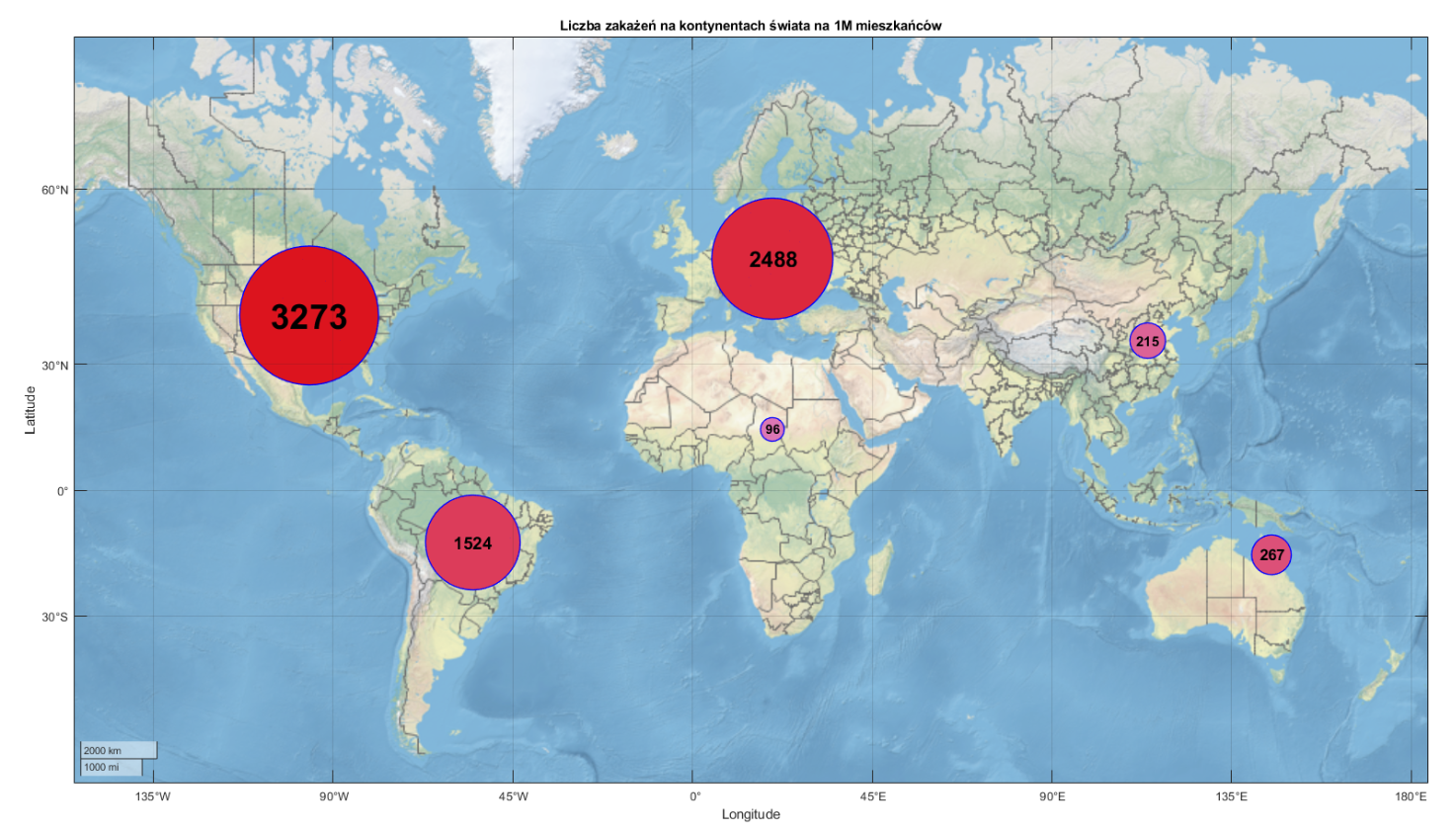


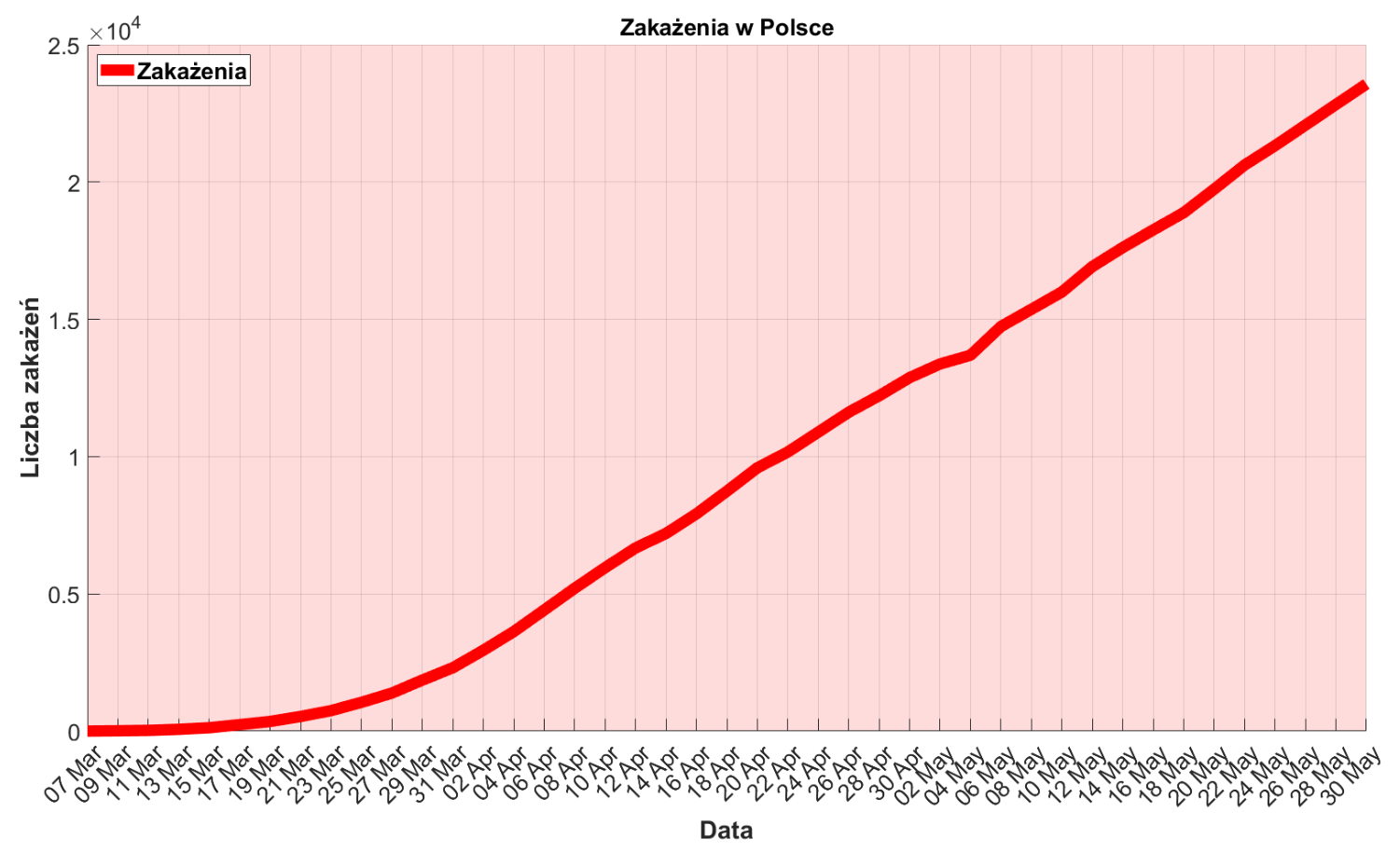












[**ANIMACJA (Obraz .gif)**](https://s7.gifyu.com/images/poland_dbd.gif)



