



Analizy epidemiologiczne w środowisku MATLAB/Octave

Agnieszka Bartłomiejczyk, Marcin Wata

Gdańsk, 20 września 2019

Spotkanie akademickie na PG, marzec 2017



URZĄD MARSZAŁKOWSKI
WOJEWÓDZTWA POMORSKIEGO

Unia Europejska
Europejski Fundusz Społeczny



Politechnika Gdańska

Spotkanie akademickie: **25 marca 2017 r. (sobota), godz. 10:00**

Dla uczniów z przedmiotu: **matematyka**

Poziom: **ponadgimnazjalny**

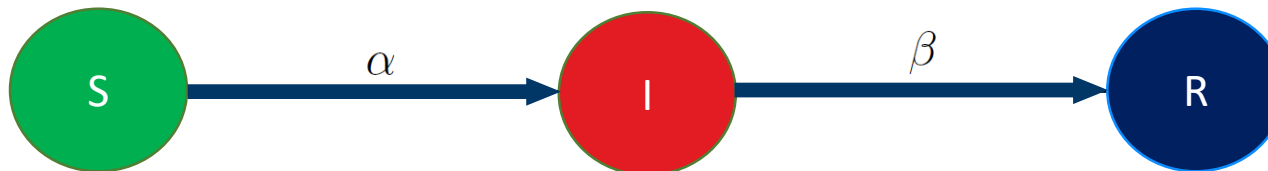


Temat: **Czy matematyka może zwalczyć epidemie?**

Model SIR (model Kermacka-McKendricka)

Populację dzielimy na trzy grupy, tj.

- S – osobniki podatne na infekcję, które nie mają odporności (*susceptible*),
- I – osobniki zainfekowane, które w wyniku kontaktu z innymi mogą przekazywać chorobę (*infected*),
- R – osobniki, które nabyły odporność na daną infekcję i nie zarażają (*resistant*).



Model SIR (model Kermacka-McKendricka)

Matematyczny opis dynamiki zmian liczby zachorowań

$$\begin{aligned}\frac{dS}{dt} &= -\alpha I(t)S(t) \\ \frac{dI}{dt} &= \alpha I(t)S(t) - \beta I(t) \\ \frac{dR}{dt} &= \beta I(t),\end{aligned}$$

gdzie α , oznacza średnią liczbę kontaktów powodujących infekcję oraz β jest współczynnikiem wyzdrowień (czyli przejścia osobników zainfekowanych po okresie choroby do grupy osobników odpornych na daną infekcję).

Model SIR (model Kermacka-McKendricka)

fun_sir.m

```
function dw = fun_sir(t,w)
global alpha beta;

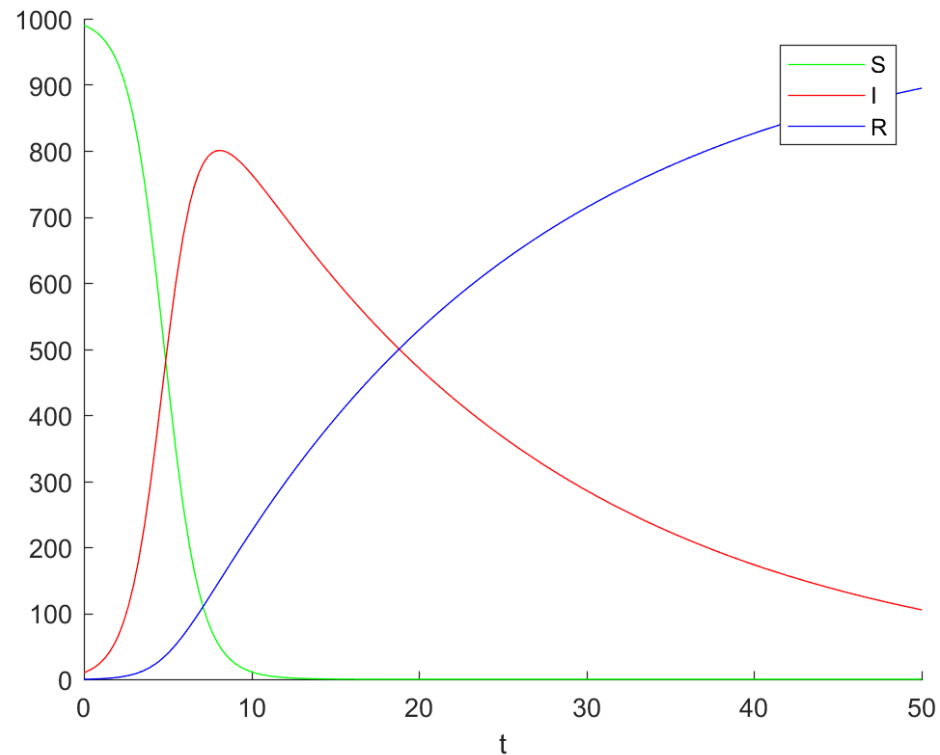
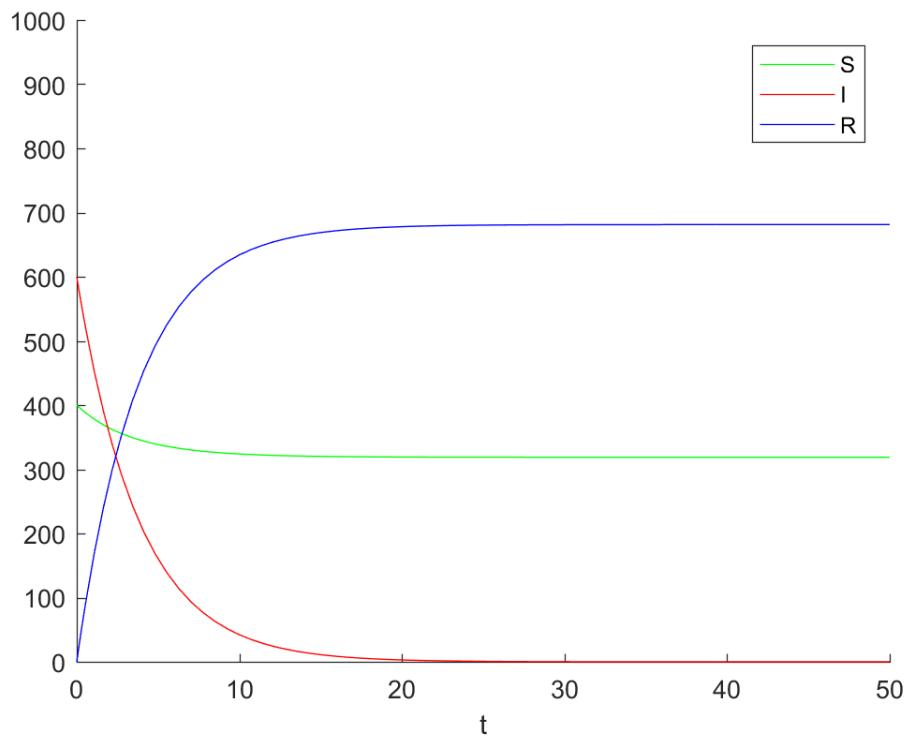
alpha = 0.1*2/1000;
beta = 0.1;

dw = zeros(3,1);
dw(1) = -alpha*w(1)*w(2);
dw(2) = alpha*w(1)*w(2)-beta*w(2);
dw(3) = beta*w(2);
end
```

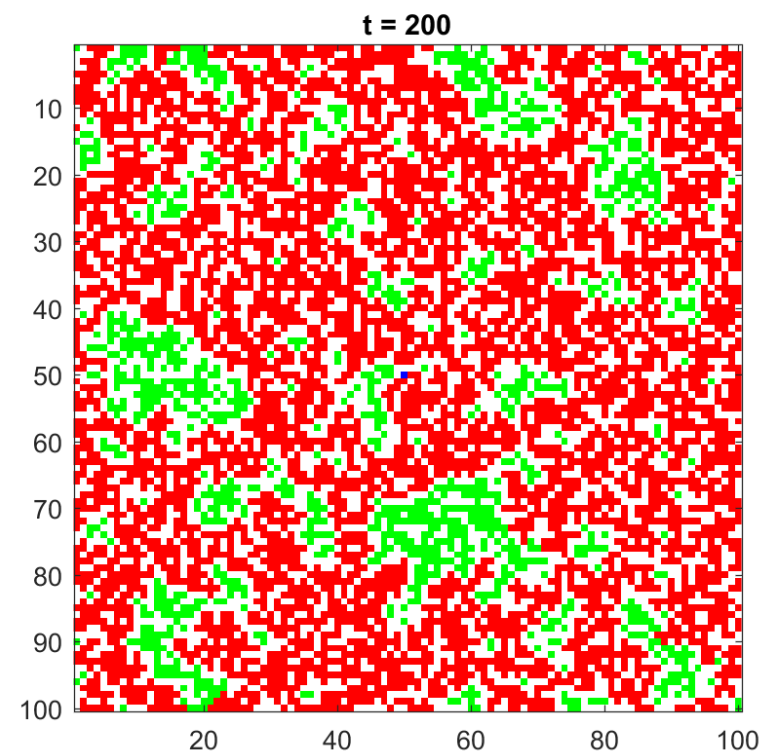
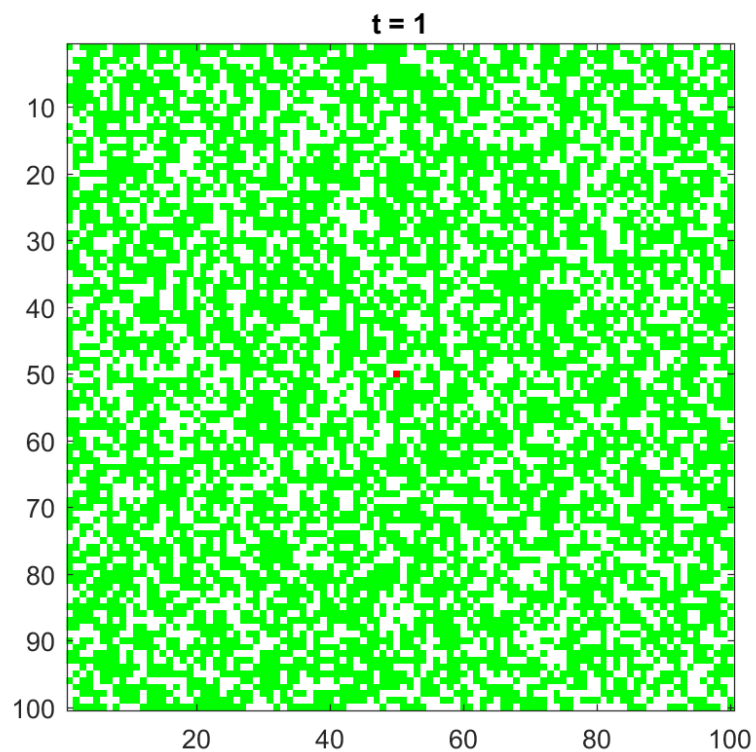
sir.m

```
% Numeryczne rozwiązanie układu równań różniczkowych w przedziale
[0,T]
% dla warunku początkowego [S,I,R]
T = 150; % czas
N = 1000;% populacja
I = 50; %populacja osobników zainfekowanych
S = N-I; % populacja osobników podatnych
R = 0; % populacja osobników odpornych
[t,y] = ode45('fun_sir',[0, T],[S,I,R]);
figure
plot(t,y(:,1),'green',t,y(:,2),'red',t,y(:,3),'blue');
xlabel('t');
legend('S','I','R');
```

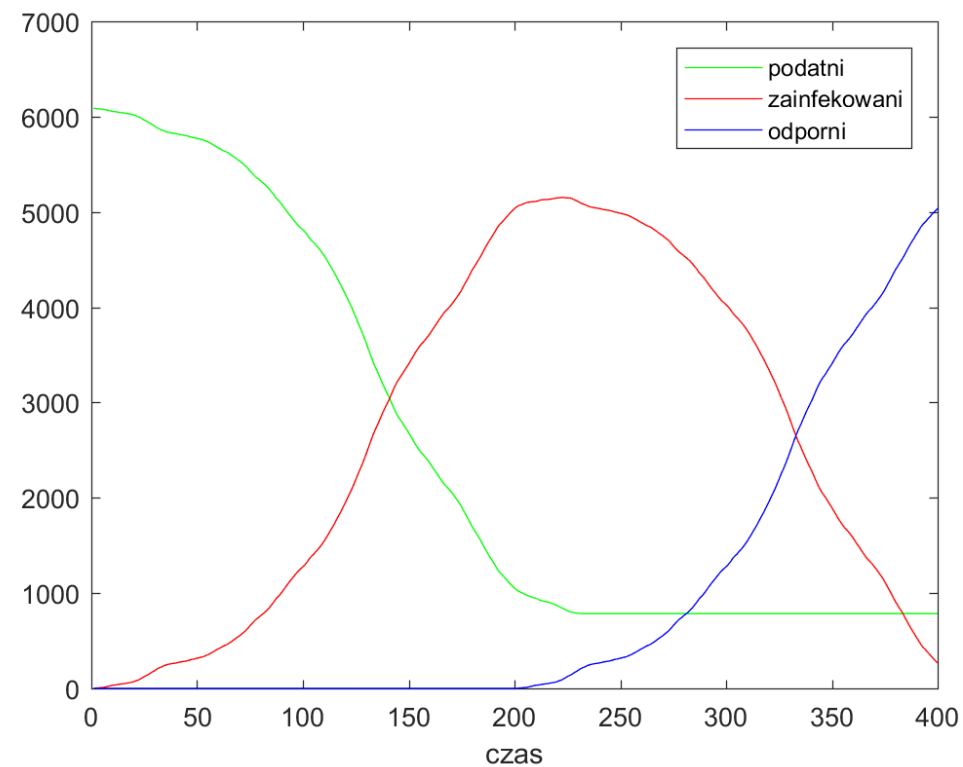
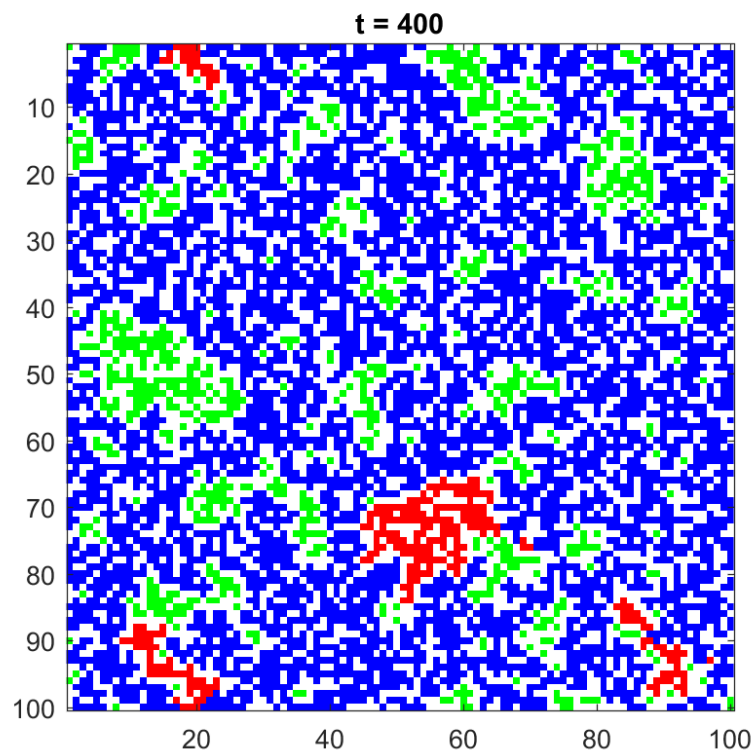
Model SIR (model Kermacka-McKendricka)



Model SIR – wykorzystanie automatów komórkowych



Model SIR – wykorzystanie automatów komórkowych



MATLAB/Octave – kursy MOOC

coursera Explore

What do you want to learn?



Introduction to Programming with MATLAB > Week 1 > Syllabus

The course is based on a video series developed at Vanderbilt University, named Computer Programming with MATLAB. It is based on an [electronic textbook](#) by the same authors which is a recommended reading for this course. You will find the book invaluable in understanding MATLAB deeper and it will also help with the programming assignments. The book also covers more advanced features of MATLAB such as Object Oriented Programming and Graphical User Interfaces that this MOOC does not get into. The course is divided into the following 8 weekly modules:

Lesson 1: Introduction to MATLAB

We will learn how to start MATLAB and will familiarize ourselves with its user interface. We will learn how to use MATLAB as a sophisticated calculator. We will learn about syntax and semantics. We will see ways in which MATLAB provides help. Finally, we will learn how to create plots in MATLAB.

Lesson 2: Matrices and Operators

The basic unit with which we work in MATLAB is the matrix. We solve problems by manipulating matrices, and operators are the primary means by which we manipulate them. We will learn how to define matrices, extract parts of them and combine them to form new matrices. We will learn how to use operators to add, subtract, multiply, and divide matrices, and we will learn that there are several different types of multiplication and division. Finally, we will learn MATLAB's rules for determining the order in which operators are carried out when more than one of them appear in the same expression.

Lesson 3: Functions

Functions let us break up complex problems into smaller, more manageable parts. We will learn how functions let us create reusable software components that can be applied in many different programs. We will learn how the environment inside a function is separated from the outside via a well defined interface through which it communicates with that outside world. We will learn how to define a function to allow input to it when it initiates its execution and output from it when it is done.

Lesson 4: Programmer's Toolbox

MATLAB has useful built-in functions and we will explore many of them in this section. We will learn about polymorphism and how MATLAB exploits it to change a function's behavior on the basis of the number and type of its inputs. Because random numbers play an important role in computer programming, we will learn how to use the MATLAB random number generator. We will learn how to get input from the keyboard, how to print to the Command Window, and how to plot graphs in a Figure window. Finally, we will learn how to find programming errors with the help of the debugger.

edX TUMx: MATLABx
Einführung in MATLAB







Course Discussion Wiki Progress

Einführung in MATLAB

- Einführung
- 2. Vektoren und Matrizen
- 3. Plotten
- 4. Skripten und Funktionen
- 5. Programmieren in MATLAB
- 6. Komplexe Zahlen
- 7. Nullstellenbestimmung
- ▼ 8. Numerisches Differenzieren und Integrieren
 - ▼ Numerisches Differenzieren und Integrieren Quizzes
- Inhalt
- 9. Anfangswertprobleme

MATLAB/Octave na zajęciach z Matematyki

Octave - wprowadzenie

-  [Instalacja Octave i pierwsze kroki](#)
-  [Macierze w Octave](#)
-  [Instrukcja warunkowa](#)
-  [Pętle w Octave](#)
-  [Funkcje w Octave](#)
-  [Nasza pierwsza poważniejsza funkcja - wzory Cramera](#)

Algorytm eliminacji Gaussa - Octave

-  [Eliminacja Gaussa-Jordana - Przykład 1](#)

Eliminacja Gaussa-Jordana - Przykład 1



Rozważaliśmy już algorytm wyznaczania rozwiązania układu dwóch równań z dwiema niewiadomymi za pomocą algorytmu Gaussa-Jordana (przypadek, gdy układ ma jedno rozwiązanie).

Zapiszmy teraz nasz algorytm za pomocą skryptu Octave

gj2.m

```
function W = gj2 (A)
    # eliminacja w dół
    A(2,:) = A(2,:) - (A(2,1)/A(1,1))*A(1,:);
    # eliminacja w górę
    A(2,:) = (1/A(2,2))*A(2,:);
    A(1,:) = A(1,:) - (A(1,2)/A(2,2))*A(2,:);
    A(1,:) = (1/A(1,1))*A(1,:);
    W = A(:,3);
endfunction
```



MATLAB/Octave na zajęciach z Matematyki

★ Przykład 2 (KZ 2010/2011)

Niech $A, B \in M_{2 \times 2}(\mathbb{R})$ będą macierzami nieosobliwymi

Wyznacz macierz X z równania

$$\left(\frac{1}{2}BX^{-1}\right)^{-1} = 2A - X$$

II. Wyznacz X jeżeli

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}, B^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Rozwiązanie

$$2B^{-1} + I = 2 \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}$$

Powyższa macierz jest odwracalna

$$(2B^{-1} + I)^{-1} = \frac{1}{-1} \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 2 & -3 \end{bmatrix}$$

$$X = 2A(2B^{-1} + I)^{-1} = \begin{bmatrix} 2 & 4 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -1 & 2 \\ 2 & -3 \end{bmatrix} = \begin{bmatrix} 6 & -8 \\ 8 & -12 \end{bmatrix}$$

$$X = \begin{bmatrix} 6 & -8 \\ 8 & -12 \end{bmatrix}$$

☐ MATLAB

Sprawdź poprawność obliczeń w MATLABIE (Octave)

W oknie poleceń wpisz:

```
>>A=[1 2; 0 2]
```

```
>>BI=[1 1; 1 0]
```

```
>>X=2*A*(2*BI+eye(2))^-1
```

Macierz *identycznościową* uzyskujemy w MATLABIE za pomocą polecenia `eye()`. Macierz *identycznościową stopnia 2* uzyskamy za pomocą polecenia `eye(2)`.

MATLAB/Octave na zajęciach z Matematyki

★ Przykład 2

Wyznaczyć punkt P_S symetryczny do punktu $P(1, 2, -2)$ względem płaszczyzny π o równaniu $x - 2y + 3z - 5 = 0$.

Rozwiązanie

Szukany punkt P_S leży na prostej prostopadłej l do płaszczyzny π przechodzącej przez punkt P . Zauważmy, że punkt przecięcia prostej l i płaszczyzny π P_R (rzut punktu P na płaszczyznę π) jest środkiem odcinka $\overline{PP_S}$. Równoważnie $\overrightarrow{PP_S} = 2 \cdot \overrightarrow{PP_R}$.

Wektor normalny $n = [1, -2, 3]$ płaszczyzny π jest jednocześnie wektorem kierunkowym prostej l .

Równanie parametryczne prostej l ma zatem postać

$$l : \begin{cases} x = 1 + t; \\ y = 2 - 2t; \\ z = -2 + 3t; \end{cases} \quad t \in \mathbb{R}$$

Współrzędne punktu P_R są zatem rozwiązaniem układu równań

$$\begin{cases} x = 1 + t \\ y = 2 - 2t \\ z = -2 + 3t \\ x - 2y + 3z - 5 = 0 \end{cases}$$

$$(1 + t) - 2 \cdot (2 - 2t) + 3 \cdot (-2 + 3t) - 5 = 0$$

$$14t - 14 = 0$$

$$t = 1$$

$$P_R = (1 + 1, 2 - 2, -2 + 3) = (2, 0, 1)$$

Stąd $\overrightarrow{PP_R} = [1, -2, 3]$ (zauważmy, że $\overrightarrow{PP_R} = [t, -2t, 3t]$ dla $t = 1$)

$$P_S = P + \overrightarrow{PP_S} = P + 2 \cdot \overrightarrow{PP_R} = (1, 2, -2) + 2 \cdot [1, -2, 3] = (1, 2, -2) + [2, -4, 6] = (3, -2, 4)$$

Odp.: Szukany punkt to $P_S(3, -2, 4)$

☐ MATLAB

Rozwiązanie przykładu 2 możemy zapisać dla ogólnych danych za pomocą funkcji w MATLABie. Danymi są: wektor współczynników równania ogólnego płaszczyzny π ($Ax + By + Cz + D = 0$) oraz wektor reprezentujący punkt P dla którego mamy znaleźć punkt symetryczny względem płaszczyzny π . Zwracanym wynikiem jest wektor reprezentujący szukany punkt.

Przeanalizuj kod poniższej funkcji

```
function PS = pswpl( WP, P )
%pswpl PS=pswpl(WP, P)
% Funkcja zwraca punkt symetryczny do P względem płaszczyzny
% reprezentowanej przez WP - współczynniki równania ogólnego

A=[-WP(1) 1 0 0 P(1); -WP(2) 0 1 0 P(2); -WP(3) 0 0 1 P(3);0 WP(1:3) -WP(4)];
% A to macierz reprezentująca układ równań na punkt przecięcia prostej i
% płaszczyzny, kolejność zmiennych t, x, y, z
AR=rref(A); % AR macierz po eliminacji Gaussa-Jordana
%AR(1,5) to wartość parametru t
PR=[P(1)+AR(1,5)*WP(1), P(2)+AR(1,5)*WP(2), P(3)+AR(1,5)*WP(3)];

PS=P+2*(PR-P); % szukany punkt symetryczny

end

Wywołanie funkcji dla danych z przykładu 2

>> pswpl([1,-2,3,-5], [1,2,-2])
```

Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii

Konieczna wiedza w dziedzinie matematyki obejmuje solidną umiejętność liczenia, znajomość miar i struktur, podstawowych operacji i sposobów prezentacji matematycznej, rozumienie terminów i pojęć matematycznych, a także **świadomość pytań, na które matematyka może dać odpowiedź.**

(...)

Pozytywna postawa w matematyce opiera się na szacunku dla prawdy oraz na chęci szukania argumentów i oceniania ich zasadności.

Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii

Kompetencje te powinny umożliwiać lepsze rozumienie korzyści, ograniczeń i zagrożeń dla ogółu społeczeństwa wynikających z teorii i zastosowań naukowych oraz technologii (w odniesieniu do podejmowania decyzji, wartości, kwestii moralnych, kultury itp.).

Umiejętności obejmują rozumienie nauki jako procesu badawczego prowadzonego za pomocą konkretnych metod, w tym obserwacji i kontrolowanych eksperymentów, zdolność do wykorzystywania logicznego i racjonalnego myślenia do weryfikowania hipotez, a także gotowość do rezygnacji z własnych przekonań, jeżeli są one sprzeczne z nowymi odkryciami naukowymi. Obejmuje to zdolność do wykorzystywania i posługiwania się narzędziami i urządzeniami technicznymi oraz danymi naukowymi do osiągnięcia celu bądź podjęcia decyzji lub wyciągnięcia wniosku na podstawie dowodów. Niezbędna jest również zdolność do rozpoznania zasadniczych cech postępowania naukowego oraz zdolność przedstawiania wniosków i sposobów rozumowania, które do tych wniosków doprowadziły.

Źródło: Dziennik Urzędowy Unii Europejskiej

ZALECENIE RADY z dnia 22 maja 2018 r. w sprawie kompetencji kluczowych w procesie uczenia się przez całe życie (2018/C 189/01)



HISTORIA MĄDROŚCIĄ
PRZYSZŁOŚĆ WYZWANIEM

Dziękuję za uwagę!