Warszawa, 25.01.2019r.

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

Metody numeryczne

Zadanie 3: Rozwiązywanie równań różniczkowych

Wykonawca zadania:

Joanna Kiesiak #22

Prowadzący projekt:

dr inż. Andrzej Miękina

Spis treści

1.Wstęp

1.1 Środowisko programu

Program został wykonany w programie MATLAB R2017b. MATLAB to pakiet programowy, którego zadaniem jest wykonywanie złożonych obliczeń matematycznych i wizualizacja wyników. Podstawowym elementem konstrukcyjnym w tym programie jest macierz, a tablica fundamentalnym typem danych.

1.2 Wstęp teoretyczny

Równanie różniczkowe zwyczajne rzędu k przyjmuje postać:

$$F\left(t, u, \frac{du}{dt}, \dots, \frac{d^k u}{dt^k}\right) = 0$$

dla $u \epsilon C^k((a,b), \boldsymbol{R}^n)$ dla $F: D \to \boldsymbol{R}^n$. Dla $\frac{\partial F}{\partial y_k}(\hat{t}, \hat{y}) \neq 0$ otrzymujemy równanie:

$$\frac{d^k u}{dt^k} = f(t, u, \frac{du}{dt}, \dots, \frac{d^{k-1} u}{dt^{k-1}})$$

Którego rozwiązaniem jest $u \epsilon C^k((a,b), \boldsymbol{R}^n)$.

Powstały różne metody numeryczne służące rozwiązaniu powyższego problemu.

W metodzie otwartej Eulera dla równania postaci:

$$\boldsymbol{u}'(t) = \boldsymbol{A} \cdot \boldsymbol{u}(t) + \boldsymbol{b} \cdot e(t)$$

dla $t \epsilon [0, T]$ algorytm numeryczny wygląda następująco:

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + h \cdot [\boldsymbol{A} \cdot \boldsymbol{u}_{n-1} + \boldsymbol{b} \cdot e(t_{n-1})]$$

dla $n = 1,2, \dots$ oraz $\boldsymbol{u}_0 = \boldsymbol{u}(0)$.

Metoda zamknięta Radaua (IIA) z kolei prowadzi do powstania tablicy współczynników:

$$\begin{array}{c|cccc}
c_k & a_{1,1} & a_{1,2} & \cdots & a_{1,K} \\
c_k & a_{2,1} & a_{2,2} & \cdots & a_{2,K} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
c_k & a_{K,1} & a_{K,2} & \cdots & a_{K,K} \\
\hline
 & w_1 & w_2 & \cdots & w_K
\end{array}$$

dla której algorytm numeryczny wygląda następująco:

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + h \cdot \sum_{k=1}^{K} w_k f_k$$

gdzie:

$$f_k = \boldsymbol{A} \cdot [\boldsymbol{u}_{n-1} + h(\sum_{\kappa=1}^{K} a_{k,\kappa} f_k)]$$

2.Zadanie 1

2.1 Treść zadania

Rozwiązać równanie różniczkowe:

12 y'' = −4 y' − 3 y;   dla tϵ*[0;10], y(0)=0, y'(0)=1*

za pomocą zamkniętej metody Radau IA rzędu 5, dla której tablica współczynników Butchera ma następującą postać:

$$\begin{array}{c|ccc}
0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\
\frac{3}{5}-\frac{\sqrt{6}}{10} & \frac{1}{9} & \frac{11}{45}+\frac{7\sqrt{6}}{360} & \frac{11}{45}-\frac{43\sqrt{6}}{360} \\
\frac{3}{5}+\frac{\sqrt{6}}{10} & \frac{1}{9} & \frac{11}{45}+\frac{43\sqrt{6}}{360} & \frac{11}{45}-\frac{7\sqrt{6}}{360} \\
\hline
 & \frac{1}{9} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{4}{9}-\frac{\sqrt{6}}{36}
\end{array}$$

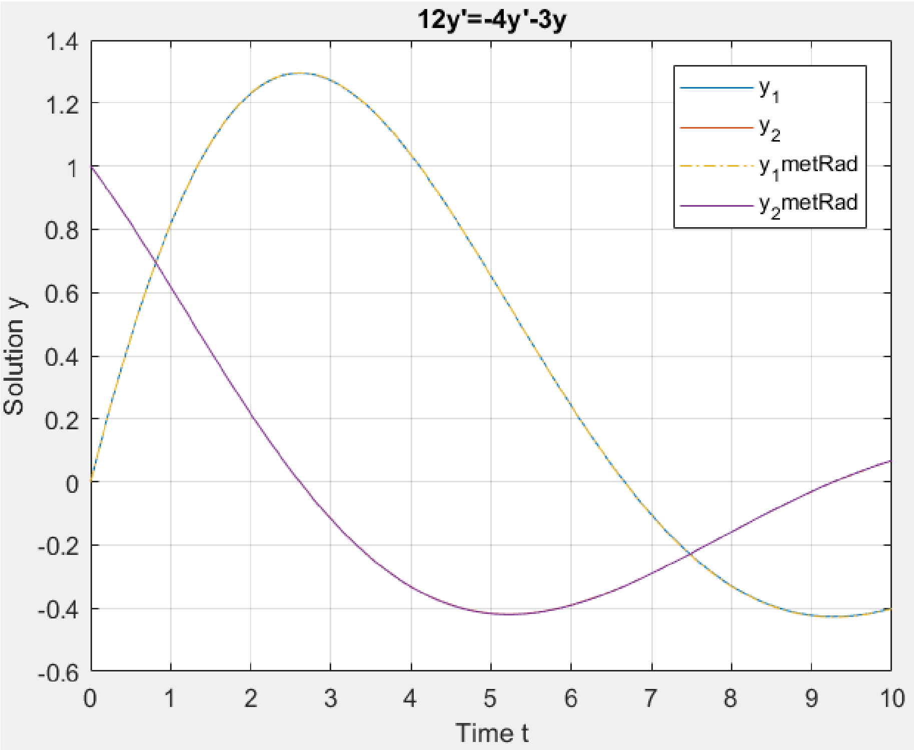
Rozwiązanie uzyskane dla stałego kroku całkowania h = 0.01 porównać z wynikiem uzyskanym za pomocą zaimplementowanej w Matlabie funkcji ***ode45****. Dobrać parametry RelTol i AbsTol tej funkcji w taki sposób, aby uzyskać możliwie dokładne rozwiązanie. Wyznaczony wektor ẏ(t) traktować jako rozwiązanie odniesienia.*

2.2 Opis zastosowanych algorytmów numerycznych

Algorytm rozwiązania równania różniczkowego składa się z dwóch metod. Pierwsza metoda wykorzystuje wbudowaną funkcję w Matlabie **ode45**. Dobrane zostały parametry RelTol oraz AbsTol na wartości odpowiednio $2e^{-2}$ oraz $1e^{-6}$. Następnie wynik jest zapisywany w argumentach T i Y i jednocześnie jest traktowany jak rozwiązanie odniesienia.

Drugie rozwiązanie jest wyznaczone za pomocą metody Radau rzędu 5 przy użyciu współczynników z tablicy Butchera. W tym celu napisano funkcję metRadau która pobiera jako argumenty wejściowe krok różniczkowania h oraz argument d, który jest ilością iteracji algorytmu oraz zwraca argumenty wyjściowe t czyli czas różniczkowania oraz y czyli estymatę rozwiązania równania według metody Radau. Następnie została zaimplementowana macierz z współczynnikami Butchera i została napisana pętla, która wylicza rozwiązanie za pomocą mnożenia macierzy, które wyznaczają estymatę. Następnie argument t i y zostały transponowane, tak aby można było je wykorzystać do wyliczenia błędu średniokwadratowego oraz błędu maksymalnego.

2.3 Przykład działania programu



rys.1 Wykres rozwiązania równania różniczkowego za pomocą dwóch metod.

3. Zadanie 2

Przeprowadzić systematyczne badania zależności dokładności uzyskanego rozwiązania od kroku całkowania h; w tym celu wyznaczyć wskaźniki błędu średniokwadratowego oraz błędu maksymalnego:

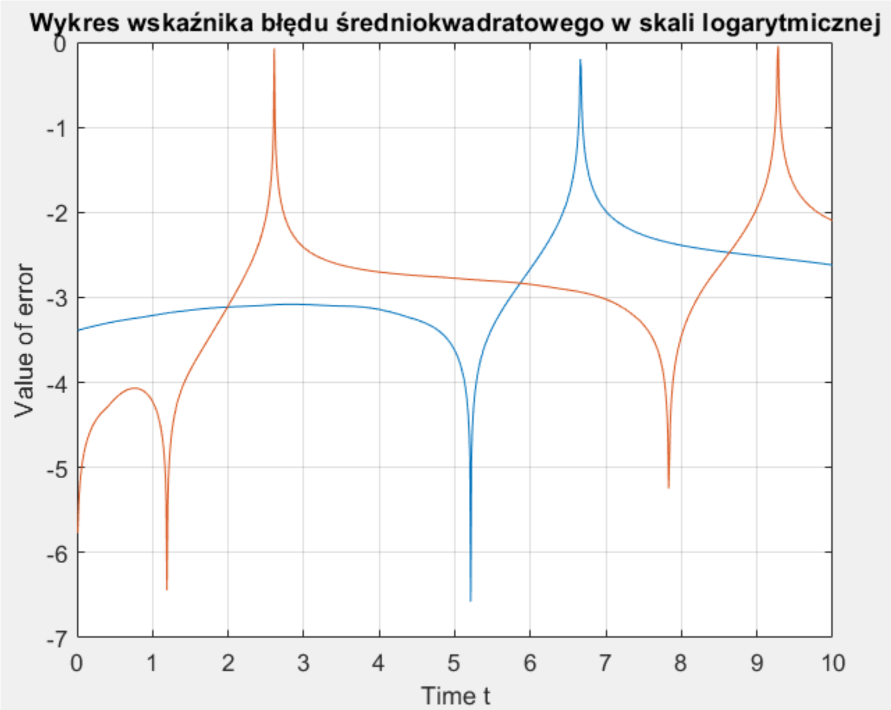
$$\delta_2(h) = \frac{\|\hat{y}(t,h) - \dot{y}(t,h)\|_2}{\|\dot{y}(t,h)\|_2} \qquad i \qquad \delta_\infty(h) = \frac{\|\hat{y}(t,h) - \dot{y}(t,h)\|_\infty}{\|\dot{y}(t,h)\|_\infty}$$

gdzie yˆ(t,h) jest rozwiązaniem uzyskanym dla kroku h, zaś ẏ(t,h) rozwiązaniem odniesienia. Sporządzić wykresy zależności $\delta_2(h)$ i $\delta_\infty(h)$.

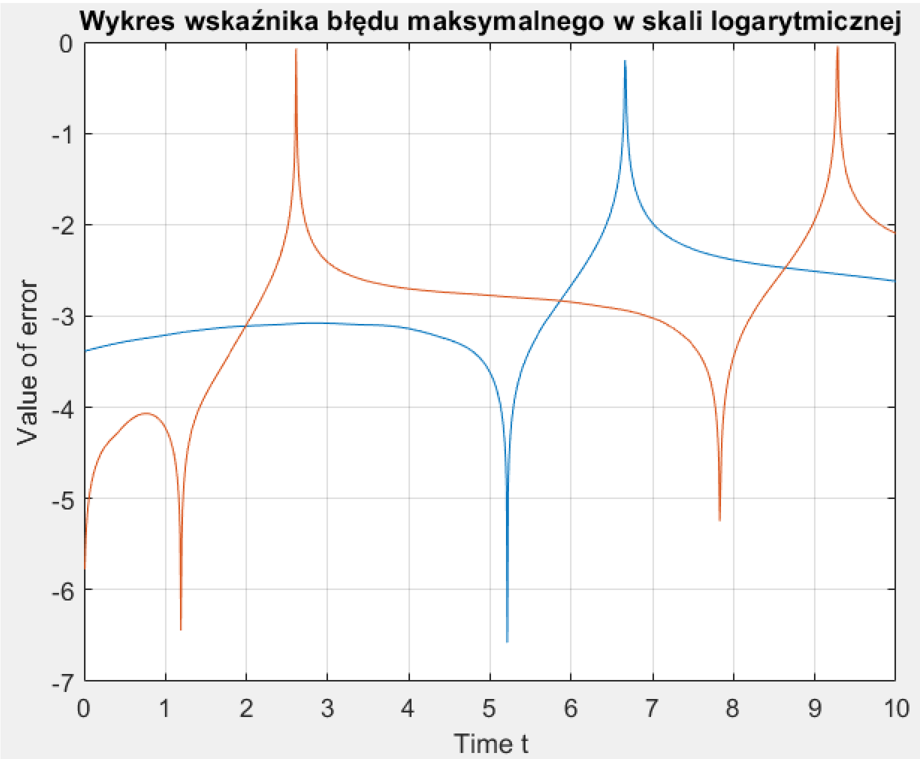
3.2 Opis zastosowanych algorytmów numerycznych

W celu wykonania zadania została napisana funkcja wyznaczBledy, która jako argumenty wejściowe pobiera dwa argumenty rozwiązania z których jedno jest rozwiązaniem odniesienia. Ta funkcja wykorzystuje funkcję norm, która jest wbudowaną funkcją w Matlabie.

3.3 Przykład działania programu



rys.2 Wykres błędu średniokwadratowego w skali logarytmicznej.



rys.3 Wykres błędu maksymalnego w skali logarytmicznej.

3.4 Wnioski

Na wykresach przedstawione są błędy rozwiązania równania różniczkowego za pomocą metody Radau, gdy rozwiązanie odniesienia to argument wyjściowy funkcji ode45. Piki, które są obecne na wykresie, skłaniają nas do wniosku że to rozwiązanie równania różniczkowego jest niestabilne i występują duże akumulacje błędów.

4.Zadanie 3

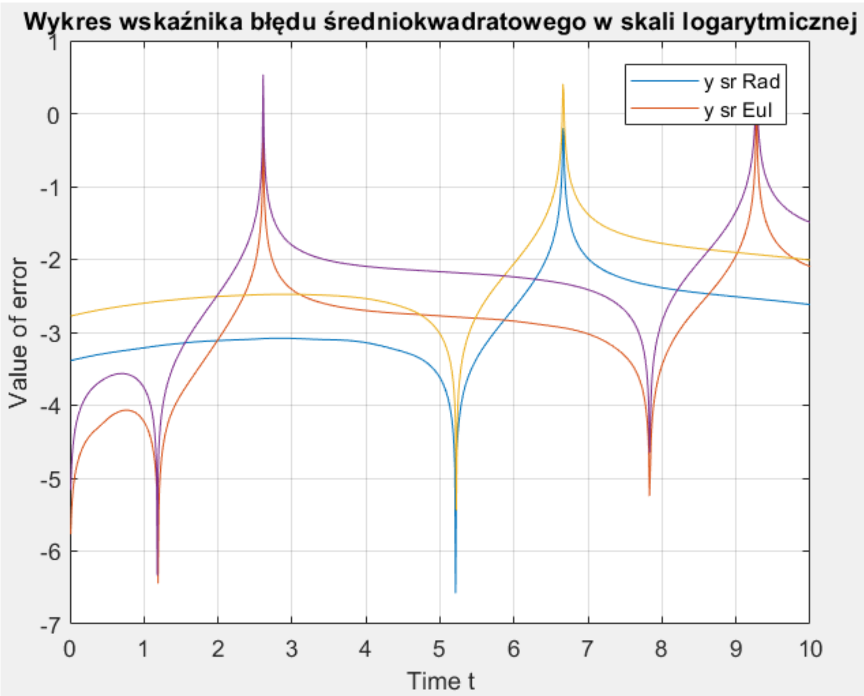
4.1 Treść zadania

Przeprowadzić analogiczne badania dla otwartej metody Eulera. Wykresy zależności $\delta_2(h)$ *i* $\delta_\infty(h)$ *dla tej metody dodać to wykresów z punktu 2.*

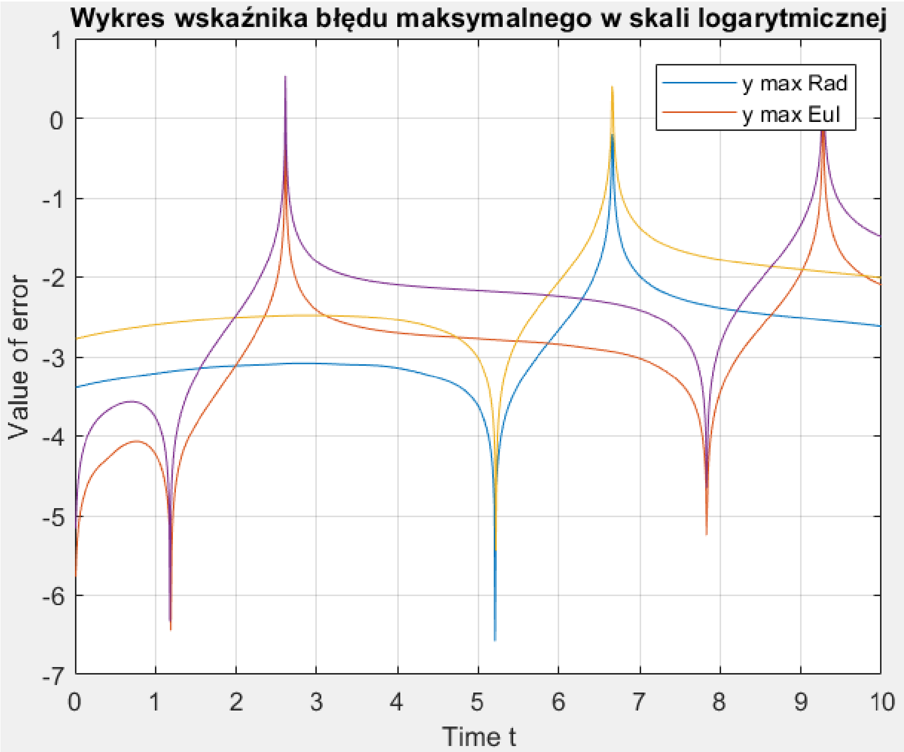
4.2 Opis zastosowanych algorytmów numerycznych

W celu wykonania zadania została napisana funkcja euler, która pobiera jako argumenty wejściowe krok różniczkowania h oraz d czyli ilość iteracji algorytmu różniczkowania, argumenty wyjściowe to czas t oraz y czyli rozwiązanie równania. Algorytm wykorzystuje znajomość kroku poprzedniego do obliczenia kroku następnego i jest realizowany za pomocą pętli for, która wyznacza nam rozwiązanie w zależności od czasu.

4.3 Przykład działania programu



rys.4 Porównanie błędu średniokwadratowego metody Radau i metody otwartej Eulera do wyznaczenia rozwiązania równania różniczkowego.



rys.5 Porównanie błędu maksymalnego metody Radau i metody otwartej Eulera do wyznaczenia rozwiązania równania różniczkowego.

4.4 Wnioski

Na wykresach porównujemy metodę otwartą Eulera i metodę Radau, obserwujemy że wykorzystanie metody Radau obniża nam błąd o ok. pół rzędu wielkości.

5. Listing

```
clear all
close all

tt = 0:0.01:10;
d = length(tt);
%rozwiązanie równania funkcją ode45
opts = odeset('RelTol',2e-2,'AbsTol',1e-6);
[T, Y] = ode45(@funk, tt, [0 1], opts);
%generacja wykresu za pomocą funkcji ode45
figure(1)
plot(T,Y(:,1),'-',T,Y(:,2))
title('12y''=-4y'-3y');
xlabel('Time t');
ylabel('Solution y');
grid on
hold on

h = 0.01;

%rozwiązanie równania za pomocą metody Radau
[Tr, Yr] = metRadau(h, d);
figure(1)
plot(Tr,Yr(:,1),'-.',Tr,Yr(:,2),'-')
legend('y_1', 'y_2','y_1metRad', 'y_2metRad');
hold on
% BLEDY
[det2r, detInfr] = wyznaczBledy(Y, Yr);
figure;
plot(tt,log10(det2r(:,1)),tt,log10(det2r(:,2)));     %rysuje wykres
title("Wykres wskaźniks błędu średniokwadratowego w skali
logarytmicznej");
xlabel('Time t');
ylabel('Value of error');
grid on;     %tworzenie linie na wykresie
hold on;
figure;
plot(tt,log10(detInfr(:,1)),tt,log10(detInfr(:,2)));     %rysuje
wykres
% set(bladMax,'ZScale','log');
title("Wykres wskaźnika błędu maksymalnego w skali
logarytmicznej");
xlabel('Time t');
ylabel('Value of error');
grid on;     %tworzenie linie na wykresie
hold on;


%rozwiązanie równania za pomocą otwartej metody Eulera
```

```
[Te, Ye] = euler(h, d);
figure(1)
plot(Te,Ye(:,1),'-',Te,Ye(:,2),'-.')
legend('y_1', 'y_2','y_1metRad', 'y_2metRad','y_1Eul', 'y_2Eul');
hold on
% BLEDY
[det2e, detInfe] = wyznaczBledy(Y, Ye);
figure;
plot(tt,log10(det2e(:,1)),tt,log10(det2e(:,2)));     %rysuje wykres
title("Wykres wskaźniks błędu średniokwadratowego w skali
logarytmicznej");
xlabel('Time t');
ylabel('Value of error');
grid on;    %tworzenie linie na wykresie
hold on;
figure;
plot(tt,log10(detInfe(:,1)),tt,log10(detInfe(:,2)));     %rysuje
wykres
% set(bladMax,'ZScale','log');
title("Wykres wskaźnika błędu maksymalnego w skali
logarytmicznej");
xlabel('Time t');
ylabel('Value of error');
grid on;      %tworzenie linie na wykresie
hold on;



function dydt = funk(t, y) %funkcja zadana
A = [0, 1; -(1/4), -(1/3)];
dydt = A*[ y(1); y(2)];
end

function [t, y] = euler(h, d)
y = zeros(2,d);
t = zeros(d,1);
%wartości początkowe
y(1,1)=0;
y(2,1)=1;
for i=1:d-1
   t(i+1) = t(i)+h;
   y(:,i+1)=y(:,i) + h*funk(t(i), y(:,i));
end
y=y';
end


function [t, y] = metRadau(h, d)
t = 0:h:10;
I = eye(2,2);
Z = zeros(2,2);
%nasza macierzA z funkcji funk
A = [0, 1; -(1/4), -(1/3)];
sqrt6=(6)^(1/2);
%współczynniki z tablicy Butchera
B = [I-(1/9)*A*h, -((-1 - sqrt6)/18)*A*h, -((-1 + sqrt6)/18)*A*h;
    (-1/9)*A*h, I-(11/45 + 7*sqrt6/360)*A*h, -(11/45 -
43*sqrt6/360)*A*h;
    (-1/9)*A*h, -(11/45 + 43*sqrt6/360)*A*h, I-(11/45 -
7*sqrt6/360)*A*h];
y = zeros(2,d);
%wartości początkowe
y(1,1) = 0;
y(2,1) = 1;
for i=1:d-1
   t(i+1)=t(i)+h;
   Ay = [A*y(:,i);A*y(:,i);A*y(:,i)];
   F = B\Ay;
   y(:,i+1)=y(:,i) + h*((1/6)*F(1:2)+(2/3)*F(3:4)+(1/6)*F(5:6));
end
t=t';
y=y';

end

function [det2, detInf] = wyznaczBledy(Y, Ywe)
[x1, x2] =size(Y);
for k =1:2
   for i = 1:x1
      det2(i,k)=norm(Ywe(i,k)-Y(i,k)) / norm(Y(i,k));
      detInf(i,k)=norm(Ywe(i,k)-Y(i,k),Inf) / norm(Y(i,k));
   end
end


end
```

6. Bibliografia

1. Morawski Roman, „Metody numeryczne (MNUB) Materiały do wykładu prowadzonego w semestrze zimowym 2017/2018”
2. Morawski Roman, „Zadania z rozwiązaniami do przedmiotów WNUM MNUB”
3. Pratap Rudra, „MATLAB 7 dla naukowców i inżynierów”, Warszawa, PWN, 2013
4. „Wstęp do metod numerycznych dla studentów elektroniki i technik informacyjnych” pod red. Romana Z. Morawskiego, Warszawa, Oficyna wydawnicza Politechniki Warszawskiej, 2009
5. https://www.mathworks.com/
6. https://www.mathworks.com/help/matlab/ref/ode45.html#bu3ujz9