METODY NUMERYCZNE

LISTA 9

WIOLETTA ŁUPKOWSKA, 244831

CZWARTEK, 9:15

1.1

a) przedstawienie równania różniczkowego drugiego rzędu w postaci dwóch równań różniczkowych pierwszego rzędu

b)

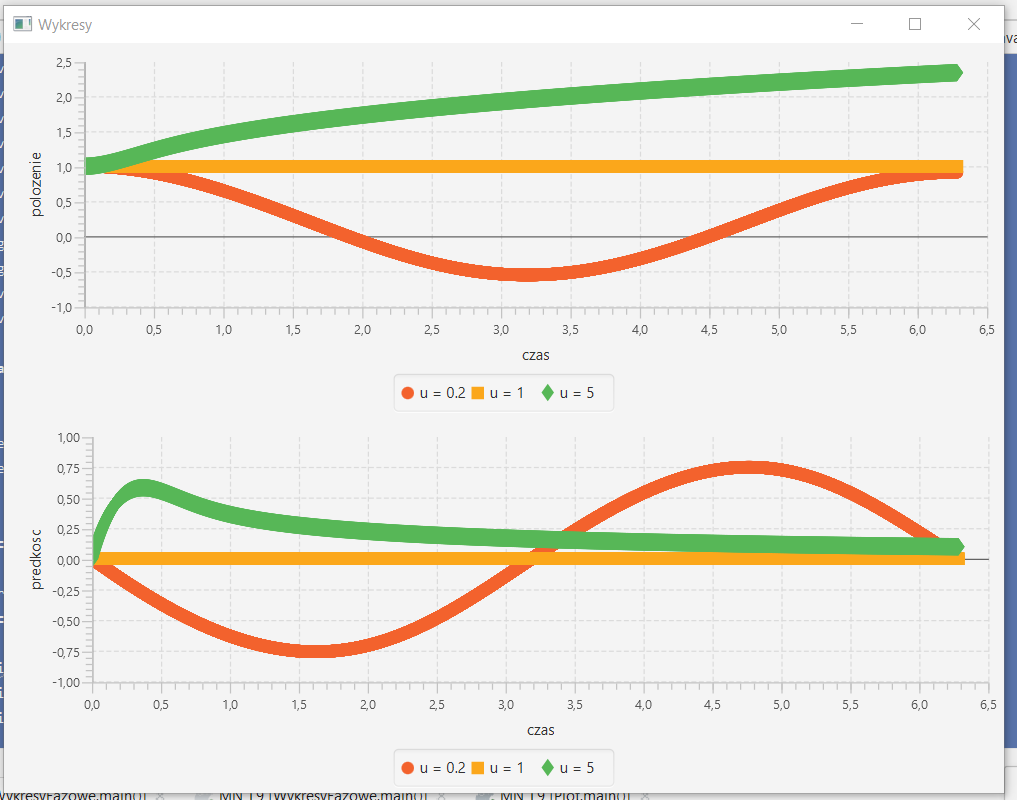
**import** org.apache.commons.math3.exception.DimensionMismatchException;  
**import** org.apache.commons.math3.exception.MaxCountExceededException;  
**import** org.apache.commons.math3.ode.FirstOrderDifferentialEquations;  
  
**public class** Function **implements** FirstOrderDifferentialEquations {  
  
 **public double u**;  
  
 **public** Function(**double** u) {  
 **this**.**u** = u;  
 }  
  
 @Override  
 **public int** getDimension() {  
 **return** 2;  
 }  
  
 @Override  
 **public void** computeDerivatives(**double** t, **double**[] x, **double**[] dxdt) **throws** MaxCountExceededException, DimensionMismatchException {  
  
 dxdt[0]= x[1]; *//tak jakby x2 z matlab* dxdt[1]=**u**\*( 1 - Math.*pow*(x[0],2) \* x[1] )- x[0];  
 }  
}

**import** org.apache.commons.math3.exception.MaxCountExceededException;  
**import** org.apache.commons.math3.ode.sampling.StepHandler;  
**import** org.apache.commons.math3.ode.sampling.StepInterpolator;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
*//HarmonicOscillatorStepHandler***public class** FStepHandler **implements** org.apache.commons.math3.ode.sampling.StepHandler {  
  
 **protected** List<Double> **tValues** = **new** ArrayList<>(); *//potem zmien na np private* **protected** List<Double> **xValues** = **new** ArrayList<>();  
 **protected** List<Double> **vValues** = **new** ArrayList<>();  
  
  
  
 @Override  
 **public void** init(**double** t0, **double**[] x0, **double** t) { *//sluzy do odczytania pocztakowych wartosci* **tValues**.add(t0);  
 **xValues**.add(x0[0]);  
 **vValues**.add(x0[1]);  
 }  
  
 @Override  
 **public void** handleStep(StepInterpolator interpolator, **boolean** isLast) **throws** MaxCountExceededException {  
  
 **double** t = interpolator.getCurrentTime();  
 **double** [] x = interpolator.getInterpolatedState();  
  
 **tValues**.add(t);  
 **xValues**.add(x[0]);  
 **vValues**.add(x[1]);  
 }  
}

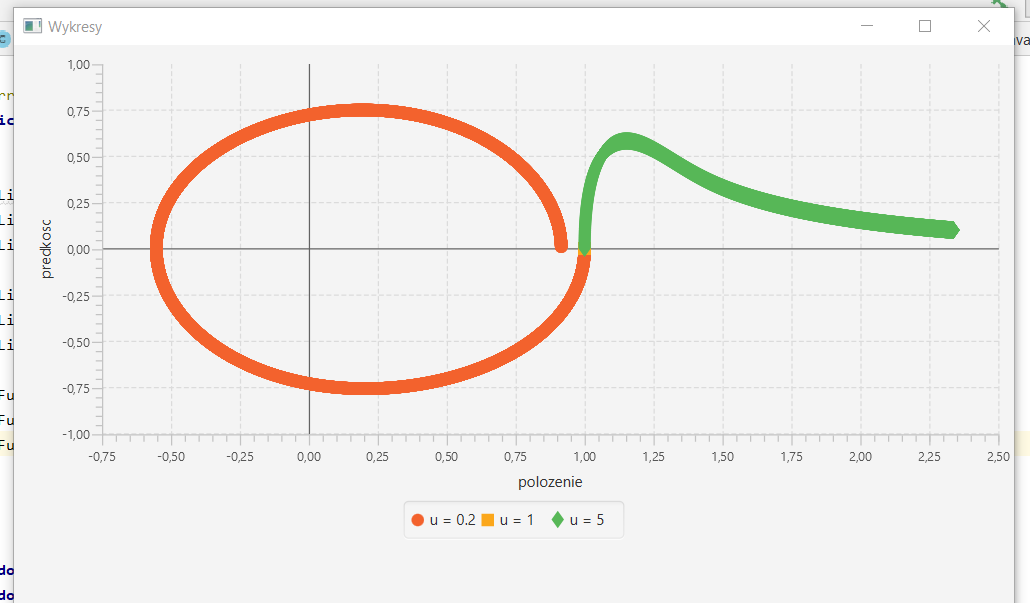
**import** javafx.application.Application;  
**import** javafx.scene.Scene;  
**import** javafx.scene.chart.NumberAxis;  
**import** javafx.scene.chart.ScatterChart;  
**import** javafx.scene.chart.XYChart;  
**import** javafx.scene.layout.VBox;  
**import** javafx.stage.Stage;  
**import** org.apache.commons.math3.ode.FirstOrderIntegrator;  
**import** org.apache.commons.math3.ode.nonstiff.EulerIntegrator;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** Plot **extends** Application {  
  
  
 ScatterChart<Number, Number> **graph**;  
 ScatterChart<Number, Number> **graphV**;  
  
  
 **public static void** main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 **public void** start(Stage primaryStage) {  
  
 List<Double> tValues02 = **new** ArrayList<>();  
 List<Double> tValues1 = **new** ArrayList<>();  
 List<Double> tValues5 = **new** ArrayList<>();  
  
 List<Double> vValues02 = **new** ArrayList<>();  
 List<Double> vValues1 = **new** ArrayList<>();  
 List<Double> vValues5 = **new** ArrayList<>();  
  
 List<Double> xValues02 = **new** ArrayList<>();  
 List<Double> xValues1 = **new** ArrayList<>();  
 List<Double> xValues5 = **new** ArrayList<>();  
  
 Function function02 = **new** Function(0.2);  
 Function function1 = **new** Function(1);  
 Function function5 = **new** Function(5);  
  
  
  
  
 **double** [] xStart = {1.,0};  
 **double** [] xStop = {0,0};  
  
 FirstOrderIntegrator eulerIntegrator = **new** EulerIntegrator(0.001);  
  
  
 FStepHandler fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function02,0.,xStart,Math.***PI***\*2, xStop);  
 tValues02= fStepHandler.**tValues**;  
 vValues02= fStepHandler.**vValues**;  
 xValues02= fStepHandler.**xValues**;  
  
 eulerIntegrator = **new** EulerIntegrator(0.001);  
 fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function1,0.,xStart,Math.***PI***\*2, xStop);  
 tValues1= fStepHandler.**tValues**;  
 vValues1= fStepHandler.**vValues**;  
 xValues1= fStepHandler.**xValues**;  
  
 eulerIntegrator = **new** EulerIntegrator(0.001);  
 fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function5,0.,xStart,Math.***PI***\*2, xStop);  
 tValues5= fStepHandler.**tValues**;  
 vValues5= fStepHandler.**vValues**;  
 xValues5= fStepHandler.**xValues**;  
  
  
  
 VBox layout= **new** VBox(); *//Layout manager  
 //utowrzenie wykresu* NumberAxis x = **new** NumberAxis();  
 x.setLabel(**"czas"**);  
 NumberAxis y = **new** NumberAxis();  
 y.setLabel(**"polozenie"**);  
 **graph** = **new** ScatterChart<>(x,y); *//number number na osiach* NumberAxis xV = **new** NumberAxis();  
 xV.setLabel(**"czas"**);  
 NumberAxis yV = **new** NumberAxis();  
 yV.setLabel(**"predkosc"**);  
 **graphV** = **new** ScatterChart<>(xV,yV); *//number number na osiach* XYChart.Series polozenie02 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series polozenie1 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series polozenie5 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series predkosc02 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series predkosc1 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series predkosc5 = **new** XYChart.Series(); *//utworzenie serii danych* **for** (**int** i=0; i<tValues02.size(); i++) {  
 polozenie02.getData().add(**new** XYChart.Data(tValues02.get(i), xValues02.get(i)));  
 predkosc02.getData().add(**new** XYChart.Data(tValues02.get(i), vValues02.get(i)));  
  
 }  
 **graph**.getData().add(polozenie02);  
 **graphV**.getData().add(predkosc02);  
*//* **for** (**int** i=0; i<tValues1.size(); i++) {  
 polozenie1.getData().add(**new** XYChart.Data(tValues1.get(i), xValues1.get(i)));  
 predkosc1.getData().add(**new** XYChart.Data(tValues1.get(i), vValues1.get(i)));  
 }  
 **graph**.getData().add(polozenie1);  
 **graphV**.getData().add(predkosc1);  
  
 **for** (**int** i=0; i<vValues5.size(); i++){  
 polozenie5.getData().add(**new** XYChart.Data(tValues5.get(i),xValues5.get(i)));  
 predkosc5.getData().add(**new** XYChart.Data(tValues5.get(i),vValues5.get(i)));  
 }  
 **graph**.getData().add(polozenie5);  
 **graphV**.getData().add(predkosc5);  
  
 polozenie02.setName(**"u = 0.2"**);  
 polozenie1.setName(**"u = 1"**);  
 polozenie5.setName(**"u = 5"**);  
 predkosc02.setName(**"u = 0.2"**);  
 predkosc1.setName(**"u = 1"**);  
 predkosc5.setName(**"u = 5"**);  
  
  
 layout.getChildren().add(**graph**);  
 layout.getChildren().add(**graphV**);  
  
 Scene scene = **new** Scene(layout, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle(**"Wykresy"**);  
 primaryStage.show();  
  
  
 }  
}

**import** javafx.application.Application;  
**import** javafx.scene.Scene;  
**import** javafx.scene.chart.NumberAxis;  
**import** javafx.scene.chart.ScatterChart;  
**import** javafx.scene.chart.XYChart;  
**import** javafx.scene.layout.VBox;  
**import** javafx.stage.Stage;  
**import** org.apache.commons.math3.ode.FirstOrderIntegrator;  
**import** org.apache.commons.math3.ode.nonstiff.EulerIntegrator;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** WykresyFazowe **extends** Application {  
  
  
 ScatterChart<Number, Number> **graphF**;  
  
 **public static void** main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 **public void** start(Stage primaryStage) {  
  
  
 List<Double> vValues02 = **new** ArrayList<>();  
 List<Double> vValues1 = **new** ArrayList<>();  
 List<Double> vValues5 = **new** ArrayList<>();  
  
 List<Double> xValues02 = **new** ArrayList<>();  
 List<Double> xValues1 = **new** ArrayList<>();  
 List<Double> xValues5 = **new** ArrayList<>();  
  
 Function function02 = **new** Function(0.2);  
 Function function1 = **new** Function(1);  
 Function function5 = **new** Function(5);  
  
  
  
  
 **double** [] xStart = {1.,0};  
 **double** [] xStop = {0,0};  
  
 FirstOrderIntegrator eulerIntegrator = **new** EulerIntegrator(0.001);  
  
  
 FStepHandler fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function02,0.,xStart,Math.***PI***\*2, xStop);  
 vValues02= fStepHandler.**vValues**;  
 xValues02= fStepHandler.**xValues**;  
  
 eulerIntegrator = **new** EulerIntegrator(0.001);  
 fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function1,0.,xStart,Math.***PI***\*2, xStop);  
 vValues1= fStepHandler.**vValues**;  
 xValues1= fStepHandler.**xValues**;  
  
 eulerIntegrator = **new** EulerIntegrator(0.001);  
 fStepHandler = **new** FStepHandler();  
 eulerIntegrator.addStepHandler(fStepHandler);  
  
 eulerIntegrator.integrate(function5,0.,xStart,Math.***PI***\*2, xStop);  
 vValues5= fStepHandler.**vValues**;  
 xValues5= fStepHandler.**xValues**;  
  
  
  
 VBox layout= **new** VBox(); *//Layout manager* NumberAxis xF = **new** NumberAxis();  
 xF.setLabel(**"polozenie"**);  
 NumberAxis yF = **new** NumberAxis();  
 yF.setLabel(**"predkosc"**);  
 **graphF** = **new** ScatterChart<>(xF,yF);  
  
  
 XYChart.Series faza02 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series faza1 = **new** XYChart.Series(); *//utworzenie serii danych* XYChart.Series faza5 = **new** XYChart.Series(); *//utworzenie serii danych* **for** (**int** i=0; i<xValues02.size(); i++) {  
 faza02.getData().add(**new** XYChart.Data(xValues02.get(i), vValues02.get(i)));  
 }  
  
 **graphF**.getData().add(faza02);  
  
 **for** (**int** i=0; i<xValues1.size(); i++) {  
 faza1.getData().add(**new** XYChart.Data(xValues1.get(i), vValues1.get(i)));  
 }  
  
 **graphF**.getData().add(faza1);  
  
 **for** (**int** i=0; i<xValues5.size(); i++){  
 faza5.getData().add(**new** XYChart.Data(xValues5.get(i),vValues5.get(i)));  
 }  
  
 **graphF**.getData().add(faza5);  
  
 faza02.setName(**"u = 0.2"**);  
 faza1.setName(**"u = 1"**);  
 faza5.setName(**"u = 5"**);  
  
 layout.getChildren().add(**graphF**);  
  
 Scene scene = **new** Scene(layout, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle(**"Wykresy"**);  
 primaryStage.show();  
  
  
 }  
}

1.2



Wykres fazowy:



Dla u>=1 wykres fazowy nie jest zbliżony do okręgu, czyli ruch nie wychodzi jako periodyczny. Dla przetestowania wyniki dla u<0:

