

LABORATORIUM NR 1Zadanie 1

$$x(t) = \hat{A}t^2 + \hat{B}t + \hat{C}$$

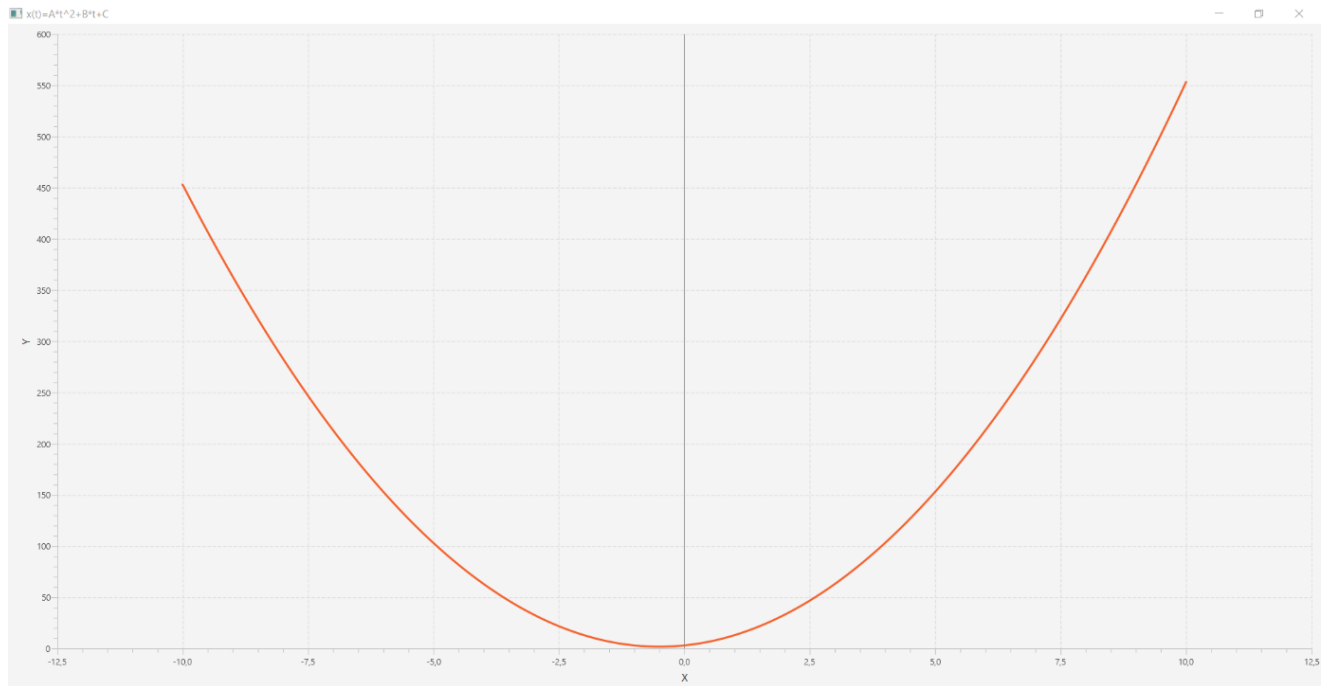
Kod programu wyliczającego wartości poszczególnych t:

```
public class ZeroOfFunction {  
  
    private static double step = 0.01;  
  
    private Map<Double, Double> scores = new HashMap<>();  
    private double a = 5;  
    private double b = 5;  
    private double c = 3;  
  
    private double countDelta() { return pow(b, 2) - 4 * a * c; }  
  
    Map<Double, Double> countScores(int start, int stop) {  
        for (double t = start; t <= stop; t += step) {  
            double tmp = a * pow(t, 2) + b * t + c;  
            scores.put(t, tmp);  
        }  
  
        return scores;  
    }  
  
    void countZerosOfFunction() {  
        double delta = countDelta();  
  
        if (delta < 0) {  
            System.out.println("There's no zeros of function.");  
        } else if (delta == 0) {  
            double x0 = -(b / 2 * a);  
            System.out.println("There's one zero of function - x0 = " + x0);  
        } else {  
            double x1 = ((-b + sqrt(delta)) / (2 * a));  
            double x2 = ((-b - sqrt(delta)) / (2 * a));  
            System.out.println("There's two zeros of function - x1 = " + x1 + "and x2 = " + x2);  
        }  
    }  
}
```

Wynik:

```
> Task :makeChart.main()  
There's no zeros of function.
```

Wykres x(t):

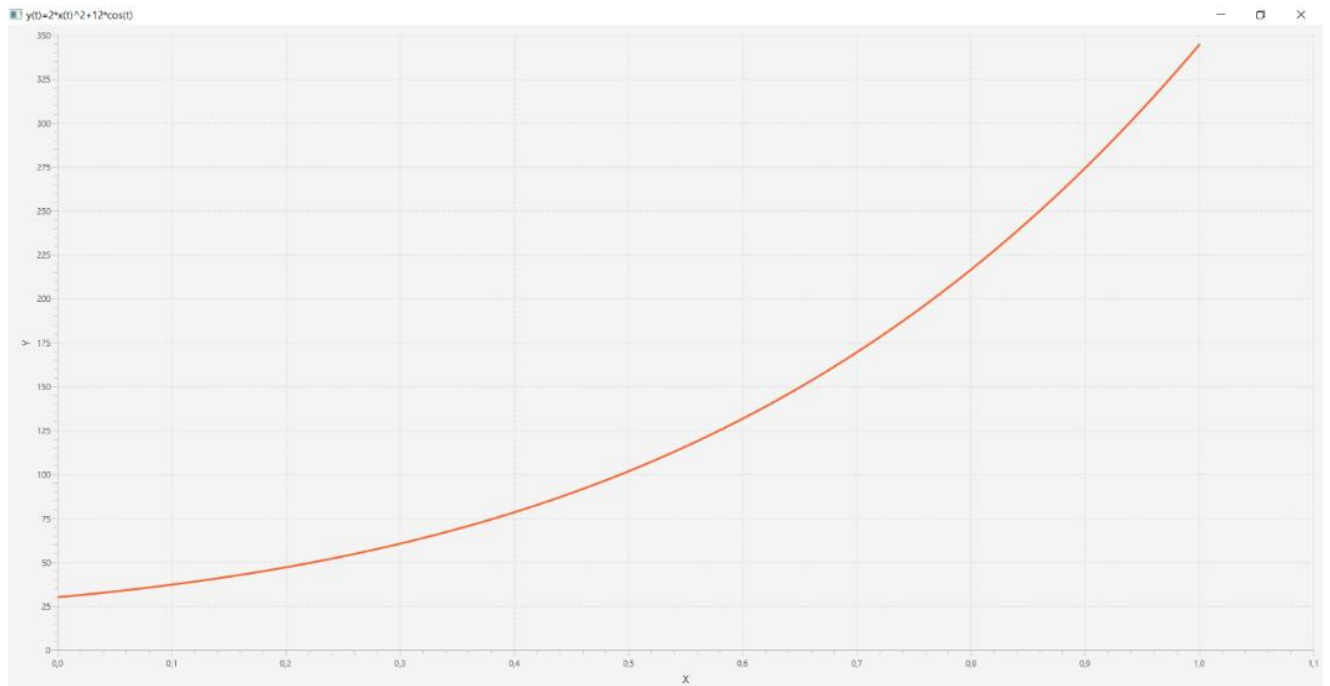
**Zadanie 2**

$$y(t) = 2 \times x(t)^2 + 12 \times \cos(t)$$

Kod programu wyliczającego wartości poszczególnych t:

```
public class Zad_2y {  
    private static double step = 0.00004535;  
    Double countX (double t) { return a*pow(t,2)+b*t+c; }  
    private Map<Double,Double> scores = new HashMap<>();  
    private double a= 5;  
    private double b = 5;  
    private double c = 3;  
  
    Map<Double, Double> countScores(int start, int stop) {  
        for (double t = start; t <= stop; t += step) {  
            double tmp = 2*pow(countX(t),2)+12*cos(t);  
            scores.put(t, tmp);  
        }  
  
        return scores;  
    }  
}
```

Wykres y(t):



$$z(t) = \sin(2\pi \times 7 \times t) \times x(t) - 0.2 \times \log_{10}(|y(t)| + \pi)$$

Kod programu wyliczającego wartości poszczególnych t:

```
public class Zad_2z {
    private static double step = 0.00004535;

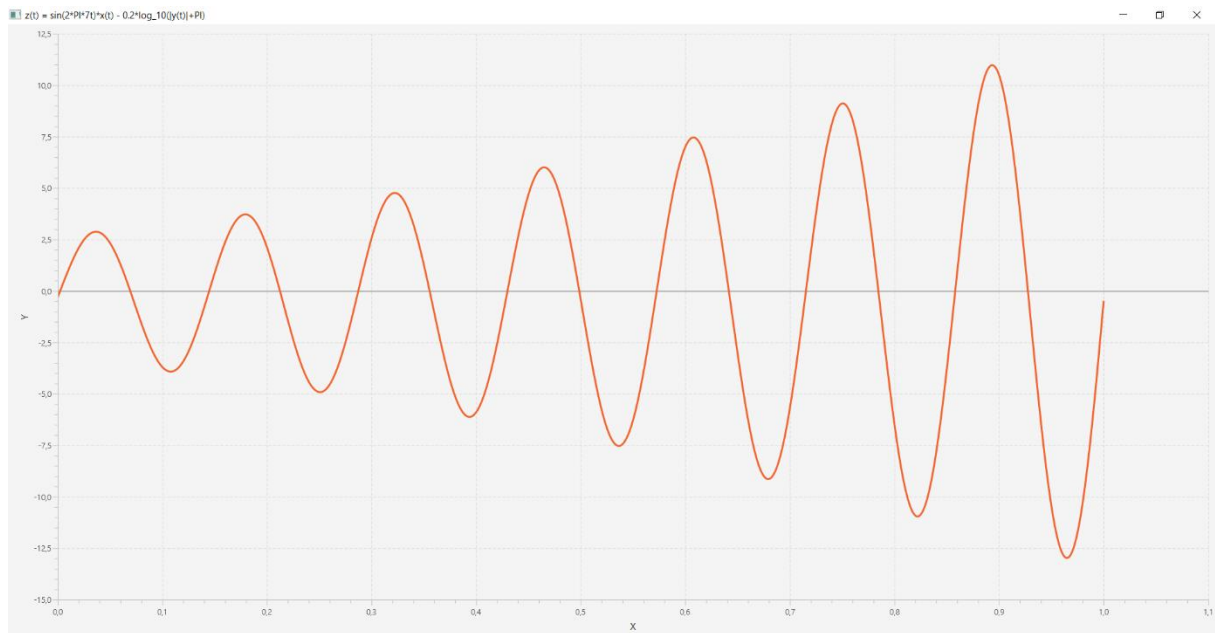
    private Map<Double, Double> scores = new HashMap<>();
    private double a = 5;
    private double b = 5;
    private double c = 3;

    Double countX (double t) { return a*pow(t,2)+b*t+c; }
    Double countY(double t) {return 2*pow(countX(t),2)+12*cos(t); }

    Map<Double, Double> countScores(int start, int stop) {
        for (double t = start; t <= stop; t += step) {
            double tmp = sin(2*PI*7*t)*(countX(t))-0.2*log10(abs(countY(t))+PI);
            scores.put(t, tmp);
        }

        return scores;
    }
}
```

Wykres z(t):



$$u(t) = \sqrt{|y(t) \times y(t) \times z(t)|} - 1.8 \times \sin(0.4 \times t \times z(t) \times x(t))$$

Kod programu wyliczającego wartości poszczególnych t:

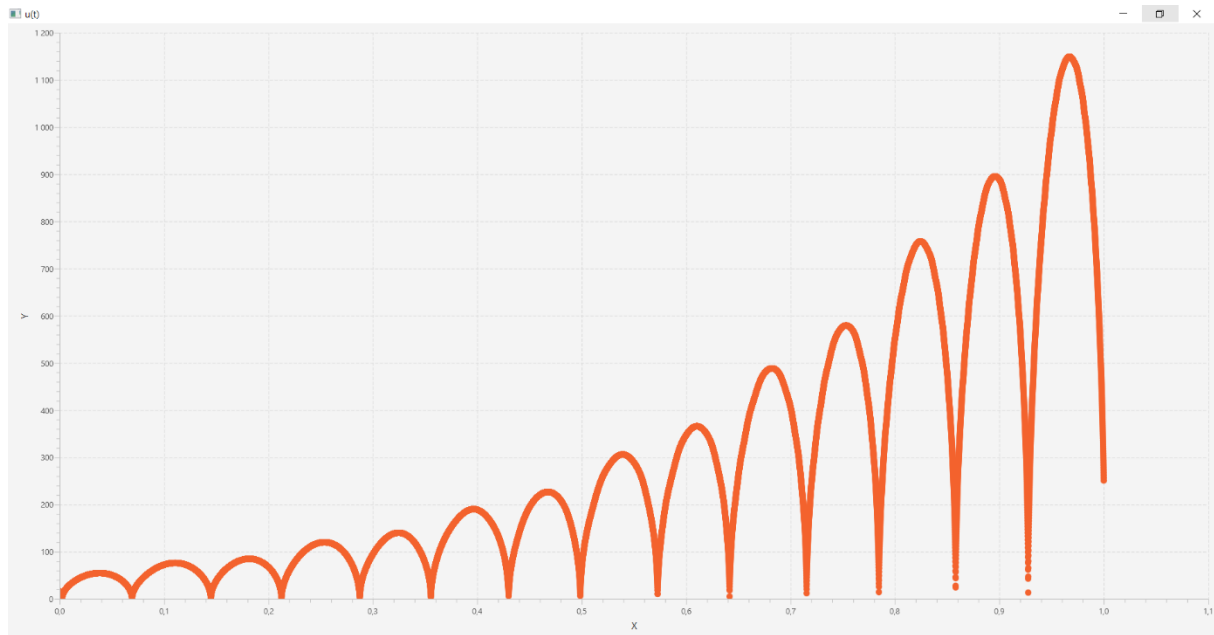
```
public class Zad_2u {
    private static double step = 0.00004535;
    private Map<Double, Double> scores = new HashMap<>();
    private double a = 5;
    private double b = 5;
    private double c = 3;

    private Double countX(double t) { return a*pow(t,2)+b*t+c; }
    private Double countY(double t) { return 2*pow(countX(t),2)+12*cos(t); }
    private Double countZ(double t) { return sin(2*PI*7*t)*(countX(t))-0.2*log10(abs(countY(t))+PI); }

    Map<Double, Double> countScores(int start, int stop) {
        for (double t = start; t <= stop; t += step) {
            double tmp = sqrt(abs(countY(t)*countY(t)*countZ(t)))-1.8*sin(0.4*t*countZ(t)*countX(t));
            scores.put(t, tmp);
        }

        return scores;
    }
}
```

Wykres u(t):



$$v(t) = \begin{cases} (1 - 7t) \times \sin\left(\frac{2\pi \times t \times 10}{t + 0.04}\right) & \text{dla } 0.22 > t \geq 0 \\ 0.63 \times t \times \sin(125 \times t) & \text{dla } 0.22 \leq t < 0.7 \\ t^{-0.662} + 0.77 \sin(8t) & \text{dla } 1.0 \geq t \geq 0.7 \end{cases}$$

Kod programu wyliczającego wartości poszczególnych t:

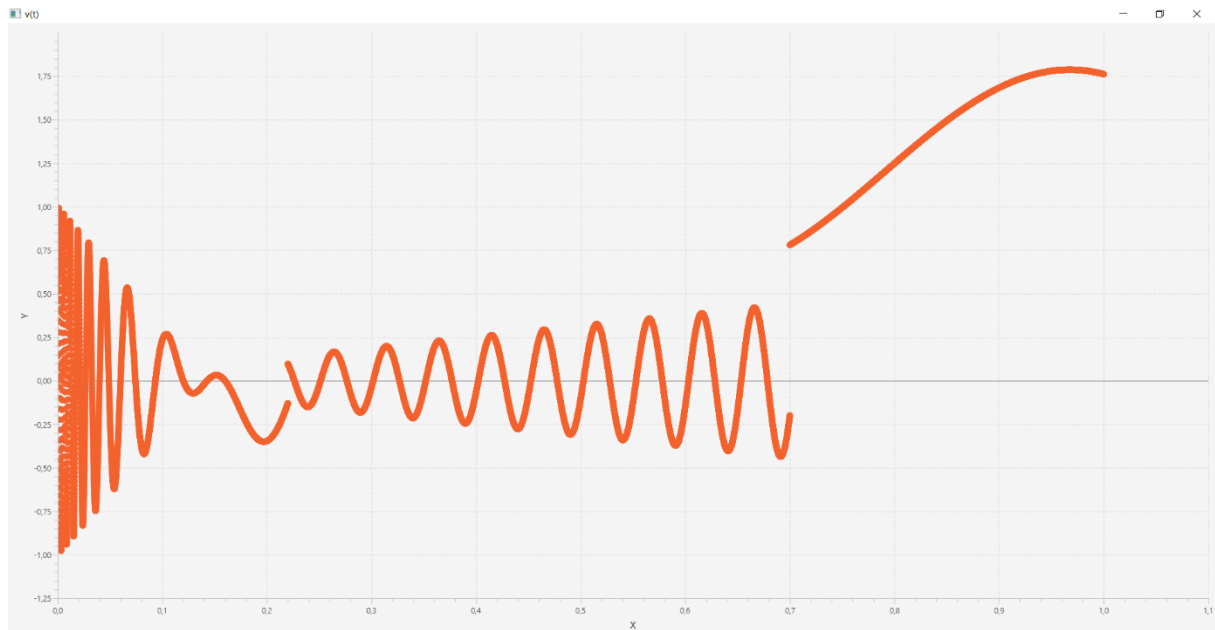
```
public class Zad_2v {
    private static double step = 0.00004535;

    private Map<Double, Double> scores = new HashMap<>();

    Map<Double, Double> countScores(int start, int stop) {
        double tmp;
        for (double t = start; t <= stop; t += step) {
            if (t >= 0 && t < 0.22)
            {
                tmp = (1 - 7 * t) * sin((2 * PI * t * 10) / (t + 0.04));
                scores.put(t, tmp);
            }
            else if (t >= 0.22 && t < 0.7)
            {
                tmp = 0.63 * t * sin(125 * t);
                scores.put(t, tmp);
            }
            else if (t <= 1 && t >= 0.7)
            {
                tmp = pow(t, -0.662) + 0.77 * sin(8 * t);
                scores.put(t, tmp);
            }
        }

        return scores;
    }
}
```

Wykres v(t):



$$p(t) = \sum_{n=1}^N \frac{\cos(12t \times n^2) + \cos(16t \times n)}{n^2} \text{ dla } N \in \{2, 4, \hat{A}\hat{B}\}$$

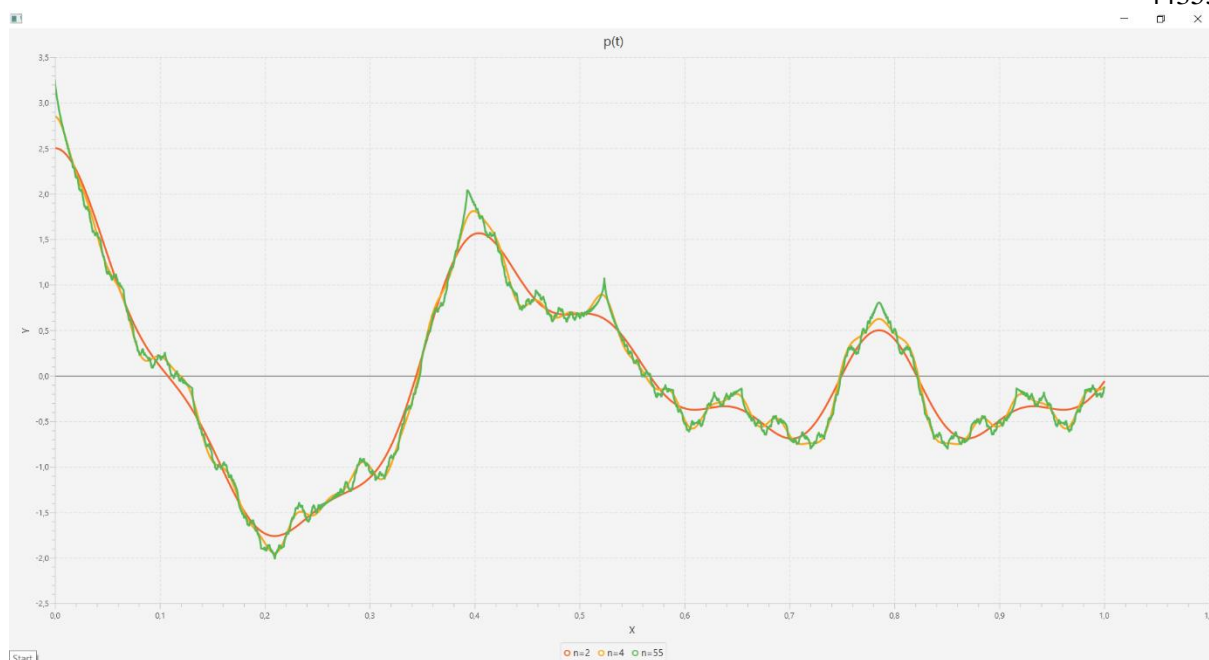
Kod programu wyliczającego wartości poszczególnych t:

```
public class Zad_2p {
    private static double step = 0.00004535;

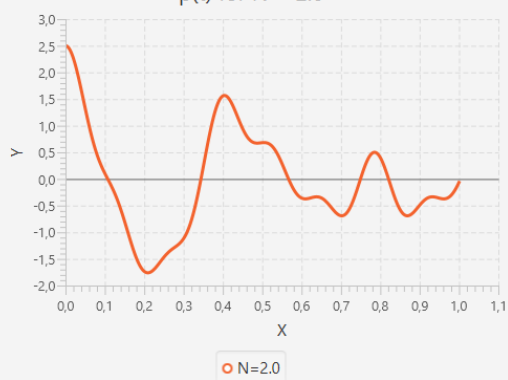
    private Map<Double, Double> scores = new HashMap<>();

    Map<Double, Double> countScores(double start, double stop, double nValue) {
        for (double t = start; t <= stop; t += step) {
            double tmp = 0;
            for (int n = 1; n <= nValue; n++) {
                tmp += (cos(12 * t * n * n) + cos(16 * t * n)) / (n * n);
            }
            scores.put(t, tmp);
        }
        return scores;
    }
}
```

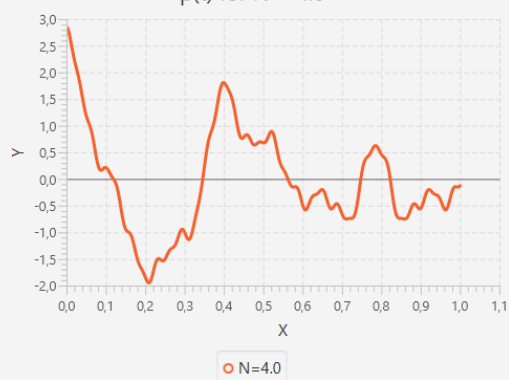
Wykres p(t):



$p(t)$ for $N = 2.0$



$p(t)$ for $N = 4.0$



$p(t)$ for $N = 55.0$

