

Laboratorium nr 1

1) Napisz procedurę/funkcję, która obliczy wyróżnik i wyznaczy miejsca zerowe zadanej funkcji kwadratowej:

$x(t) = \hat{A}t^2 + \hat{B}t + \hat{C}$. Wykonaj wykres tej funkcji dla $t \in \langle -10; 10 \rangle$, gdzie $x, t \in \mathbb{R}$, przy $\Delta t = \frac{1}{100}$.

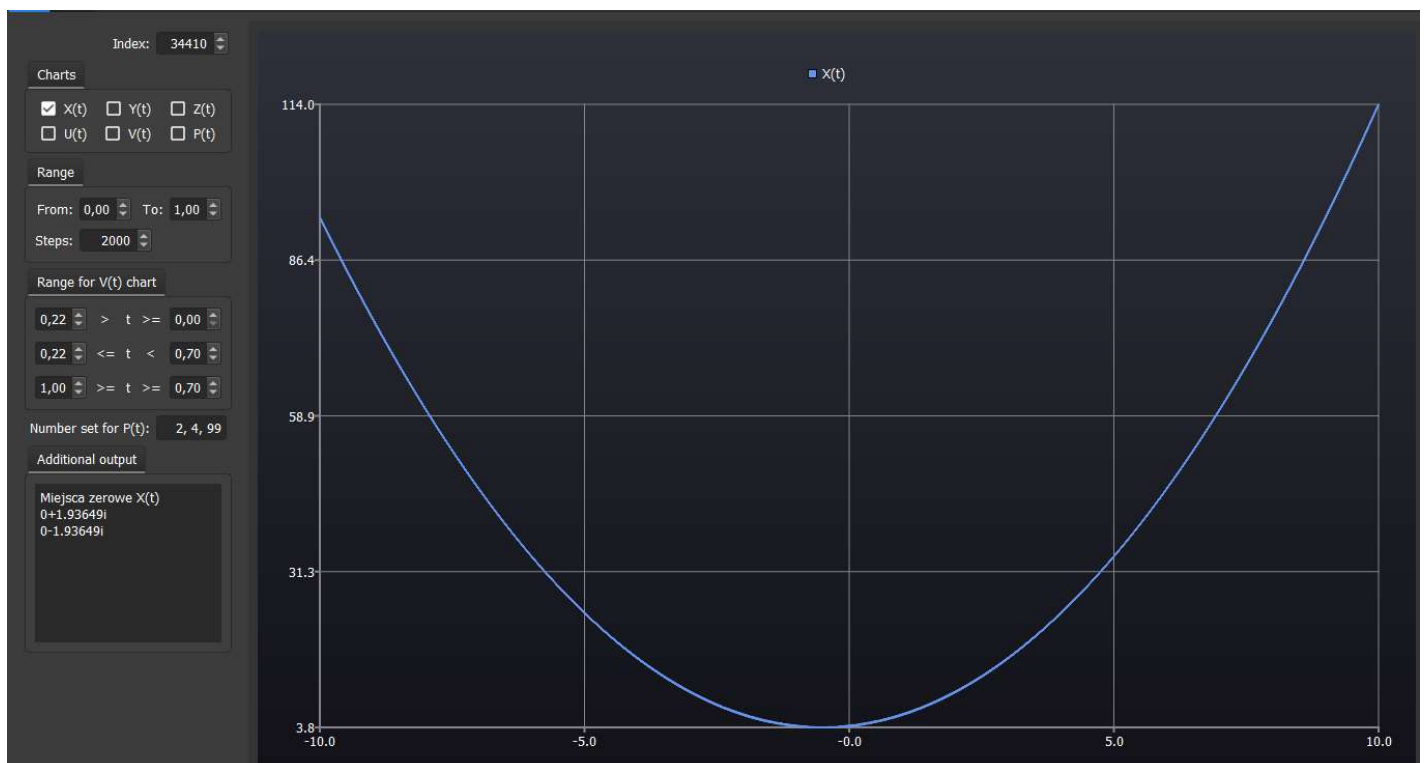
Funkcja obliczająca $x(t)$, *indexData* to zbiór cyfr w indeksie od prawej do lewej:

```
double Lab1::xFunction(double x, QVector<int> indexData) //FEDCBA
{
    Q_ASSERT(indexData.length()>=6);
    return indexData[0]*x*x+indexData[1]*x+indexData[2];
}
```

Miejsca zerowe $x(t)$:

```
double det = indexData[1]*indexData[1]-4*indexData[0]*indexData[2];
if(det>0)
{
    additionalOutput->setPlainText("Miejsca zerowe X(t)\n"+
                                   QString::number((-indexData[1]+sqrt(det))/(2*indexData[0]))
                                   +"\n"+ QString::number((-indexData[1]-sqrt(det))/(2*indexData[0])));
}
else if(det==0)
{
    additionalOutput->setPlainText("Miejsce zerowe X(t)\n"+
                                   QString::number((-indexData[1]/(2*indexData[0]))));
}
else if(det<0)
{
    additionalOutput->setPlainText("Miejsca zerowe X(t)\n"+
                                   QString::number((-indexData[1]/(2*indexData[0])))+"+"+
                                   QString::number(sqrt(-det)/(2*indexData[0]))+"i"\n"+
                                   QString::number((-indexData[1]/(2*indexData[0])))+"-"+
                                   QString::number(sqrt(-det)/(2*indexData[0]))+"i");
}
```

Wykres $x(t)$ + miejsca zerowe wynik:



2) Napisz program obliczający poniżej zadane funkcje dla $t \in \langle 0; 1 \rangle$, gdzie $\Delta t = \frac{1}{22050}$. Wykonaj wykresy tych funkcji.

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

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

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

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

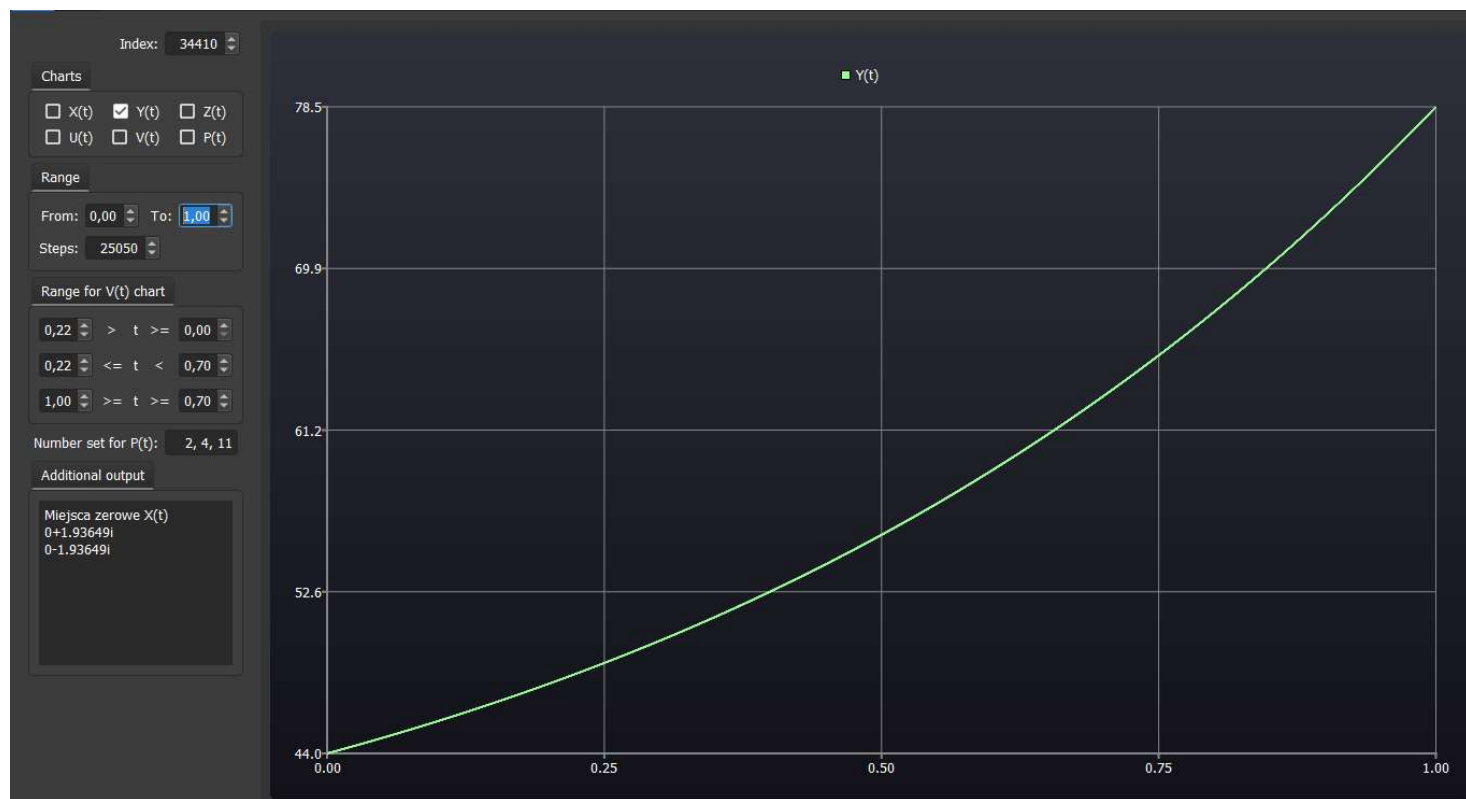
$$p(t) = \sum_{n=1}^N \frac{\cos(12t \cdot n^2) + \cos(16t \cdot n)}{n^2}$$

$$\text{dla } N \in \{2, 4, \widehat{A}\widehat{B}\}$$

Funkcje obliczające wartości $y(t)$:

```
double Lab1::yFunction(double x, QVector<int> indexData)
{
    Q_ASSERT(indexData.length() >= 6);
    return 2*(xFunction(x, indexData))*(xFunction(x, indexData)) + 12*qCos(x);
}
```

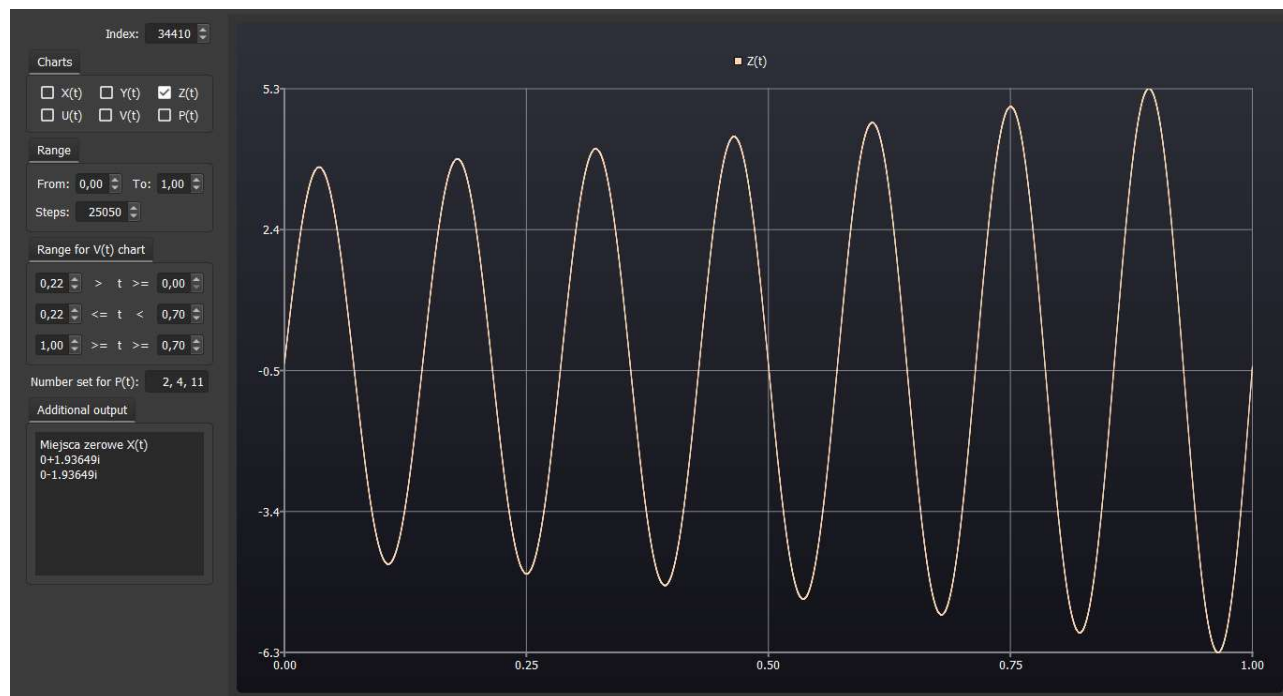
Wykres $y(t)$:



Funkcja obliczająca wartość $z(t)$:

```
double Lab1::zFunction(double x, QVector<int> indexData)
{
    Q_ASSERT(indexData.length()>=6);
    return qSin(2*M_PI*7*x)*xFunction(x, indexData) - 0.2*log10(abs(yFunction(x, indexData))+M_PI);
}
```

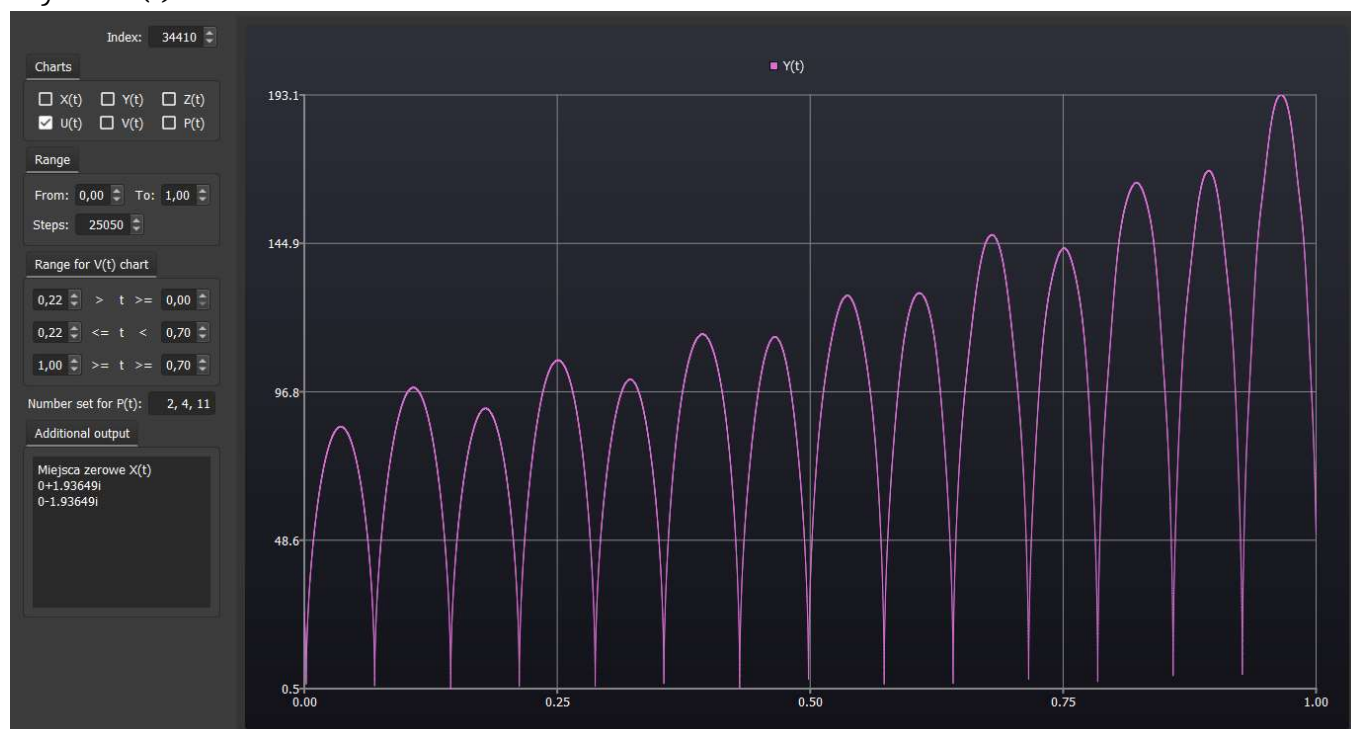
Wykres $z(t)$:



Funkcja obliczająca wartość $u(t)$:

```
double Lab1::uFunction(double x, QVector<int> indexData)
{
    Q_ASSERT(indexData.length()>=6);
    return sqrt(abs(yFunction(x, indexData)*yFunction(x, indexData)*zFunction(x, indexData)))
        - 1.8*sin(0.4*x*zFunction(x, indexData)*xFunction(x, indexData));
}
```

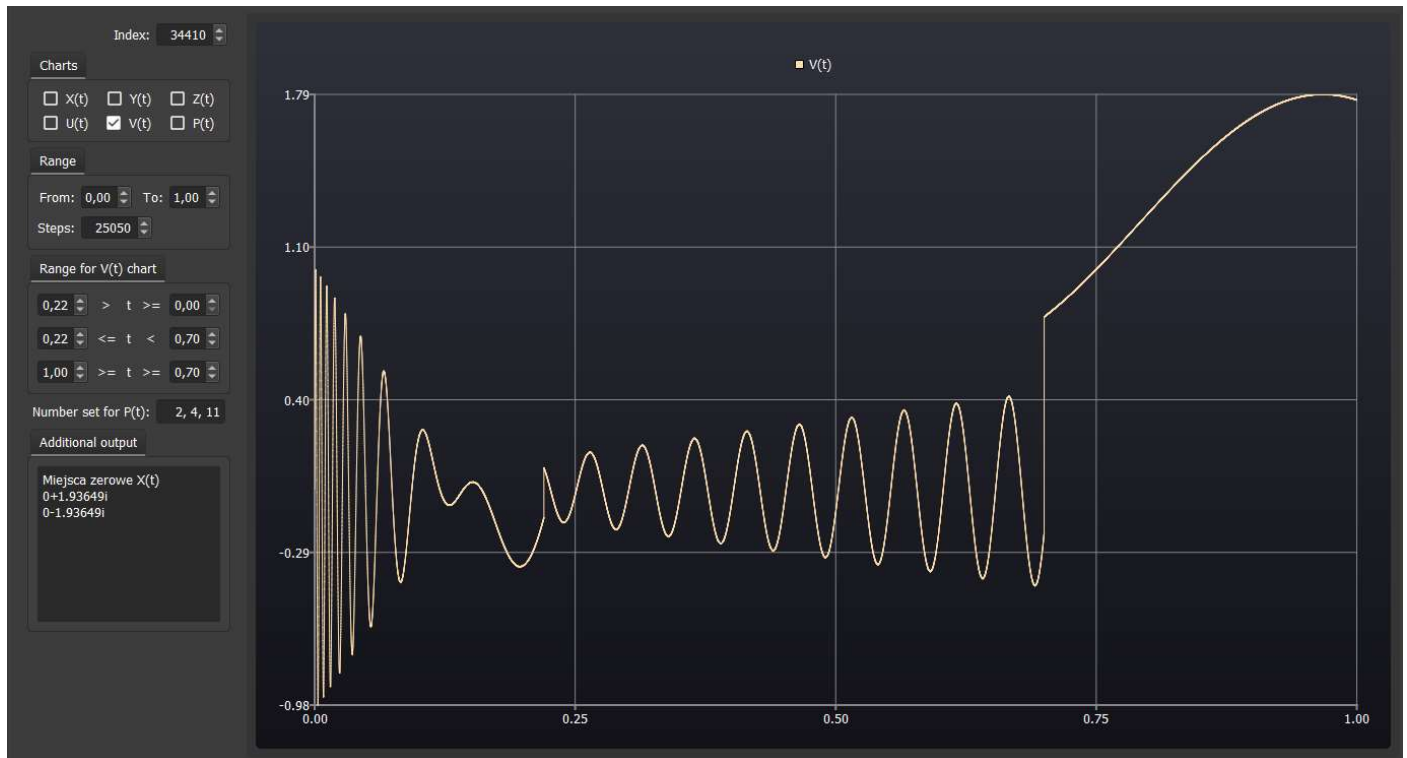
Wykres $u(t)$:



Funkcja obliczająca wartość $v(t)$:

```
double step = (range3From - range1To)/steps;  
  
for(double x = range1To; x<range1From; x+=step)  
    series->append(x, (1-7*x)*qSin((2*M_PI*x*10)/(x+0.04)));  
for(double x = range2From; x<range2To; x+=step)  
    series->append(x, 0.63*x*qSin(125*x));  
for(double x = range3To; x<=range3From; x+=step)  
    series->append(x, qPow(x, -0.662)+0.77*qSin(8*x));  
break;
```

Wykres $v(t)$:



Funkcja obliczająca wartość $p1(t)$, $p2(t)$, $p3(t)$:

```
for(double x = rangeF; x<=rangeT; x+=step)
{
    double y=0;
    for(int number=1; number<=set[0]; number++)
    {
        y += (qCos(12*x*number*number) + qCos(16*x*number))/(number*number);
    }
    series->append(x, y);
}
```

Wykresy $p1(t)$, $p2(t)$, $p3(t)$:

