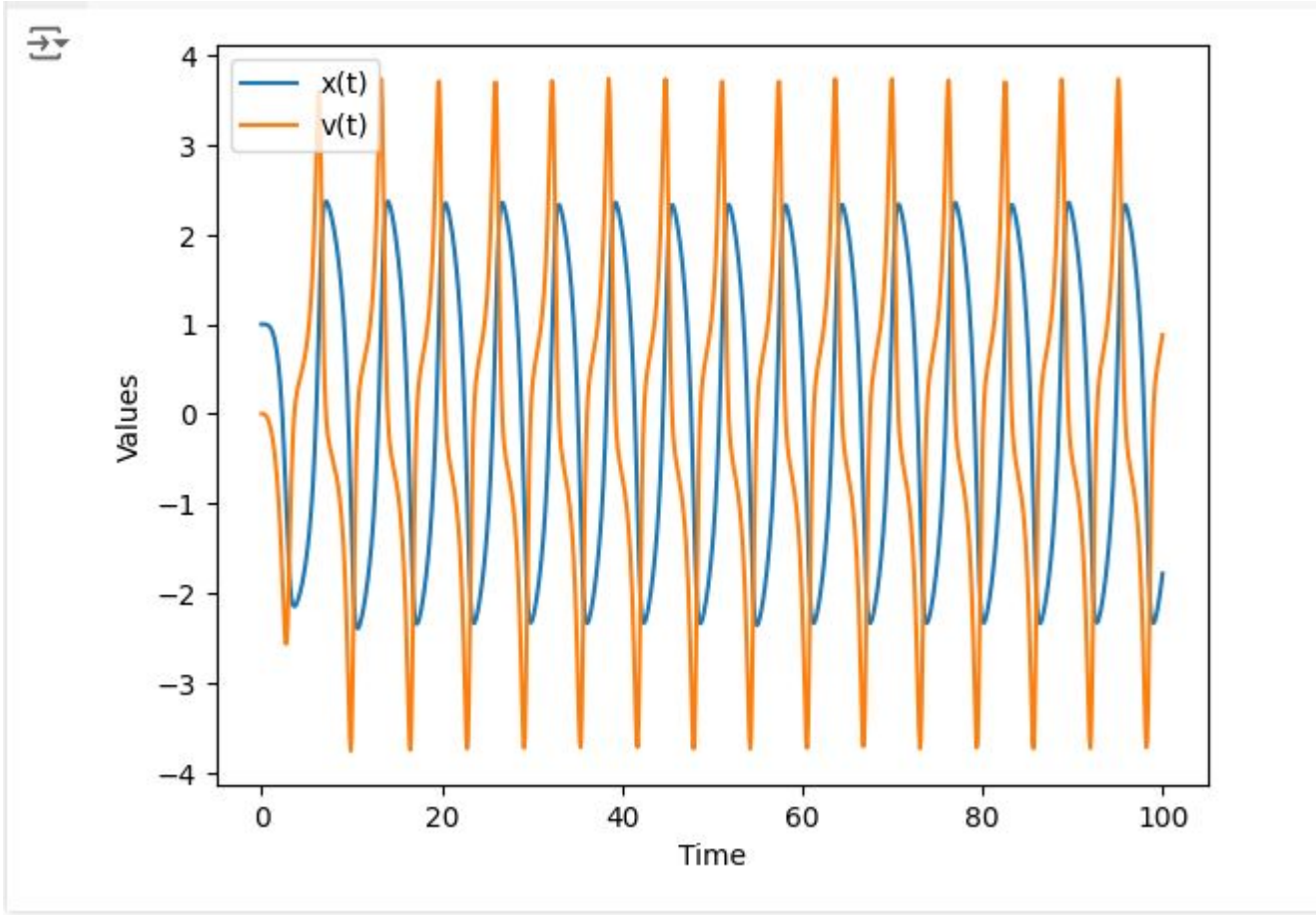
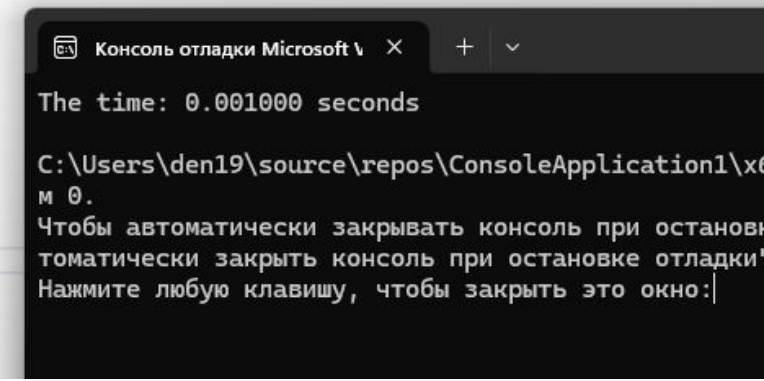
```python
import numpy as np
import math
def vander_polynomial(t, y, mu, A, w):
    x, v = y
    dydt = [v, mu*(1 - x**2)*v - x - A * math.cos(w*t)]
    return dydt

# Начальные условия
x0 = 1.0
v0 = 0.0
mu = 1.0
A = -1.0
w = -1.0

# Временной интервал
t = np.linspace(0, 100, 1000)

# Используем метод Рунге-Кутты четвертого порядка
from scipy.integrate import solve_ivp
sol = solve_ivp(vander_polynomial, [t[0], t[-1]], [x0, v0], args=(mu, A, w), t_eval=t)

# График решения
import matplotlib.pyplot as plt
plt.plot(sol.t, sol.y[0], label='x(t)')
plt.plot(sol.t, sol.y[1], label='v(t)')
plt.xlabel('Time')
plt.ylabel('Values')
plt.legend()
plt.show()
```


x(t)
v(t)
Values
Time
4
3
2
1
0
−1
−2
−3
−4
0
20
40
60
80
100

```c
int main(float* argc, char** argv)
{
    float x0 = -10.0;
    float v0 = 1.0;
    float mu = 1.0;
    float A = 20.0;
    float w = 12.0;
    float t[9000] = { 0 };
    float t1[9000] = { 0 };
    float t2[4500] = { 0 };
    float t3[4500] = { 0 };

    for (int i = 0; i < 1000; i++) {
        t[i] = i;
    }


    //clock_t start1 = clock();
        clock_t start1 = clock();
        for (int i = 0; i < 9000; i++) {
            t1[i] = vander_polynomial(x0, v0, mu, A, w, t, i);
        }
        clock_t end1 = clock();
        double seconds1 = (double)(end1 - start1) / CLOCKS_PER_SEC;
        printf("The time: %f seconds\n", seconds1);
    return 0;
}
```



```
    // printf("%.3f", vander_polynomial(0, 0, mu, A, w, t, i));
    // printf("\n");
}
MPI_Init(argc, &argv);
int size = 0, rank = 0;
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
double start, end;
MPI_Barrier(MPI_COMM_WORLD);
start = MPI_Wtime();

if (rank == 1)
{
    for (int i = 0; i < 4500; i++) {
        t2[i] = vander_polynomial(x0, v0, mu, A, w, t, i + 12);
    }
    MPI_Send(&t2, 4500, MPI_FLOAT, 0, 0, MP
}
else if (rank == 0) {
    for (int i = 0; i < 4500; i++) {
        t1[i] = vander_polynomial(x0, v0, mu, A, w, t, i);
    }
    MPI_Status status;
    MPI_Recv(&t3, 4500, MPI_FLOAT, 1, 0, MPI_COMM_WORLD, &status);
    for (int i = 0; i < 12; i++) {
      //  printf("%.3f", t1[i]);
     //   printf(" %.3f", t3[i]);
      //  printf("\n");
    }
    end = MPI_Wtime();
    printf("\nRuntime = %f\n", end - start);
}
MPI_Finalize();
return 0;
}
```

(локальная переменная) float mu

Поиск в Интернете

```
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.001000 seconds

Runtime = 0.000166
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.000000 seconds

Runtime = 0.000167
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.000000 seconds

Runtime = 0.000171
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.000000 seconds

Runtime = 0.000167
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.000000 seconds

Runtime = 0.000167
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe
The time: 0.001000 seconds

Runtime = 0.000168
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpiexec -n 2 .\ConsoleApplication1.exe

Runtime = 0.000181
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug>
```

```
if (rank == 1)
{
  #pragma omp parallel
    {
        for (int i = 5000; i < 10000; i++)
        {
            while ((pow(x[0] - x[i], 2) + pow(y[0] - y[i], 2) + pow(z[0] - z[i], 2)) > R2[i - 5000] * R2[i
            {
                R2[i - 5000] = R2[i - 5000] + 1;
            }
        }
    }
    MPI_Send(&R2, 5000, MPI_FLOAT, 0, 0, MPI_COMM_WORLD);
}
else if (rank == 0) {

 #pragma omp parallel
    {
        for (int i = 5000; i < 10000; i++)
        {
            while ((pow(x[0] - x[i], 2) + pow(y[0] - y[i], 2) + pow(z[0] - z[i], 2)) > R1[i] * R1[i])
            {
                R1[i] = R1[i] + 1;
            }
        }
    }
    MPI_Status status;
    MPI_Recv(&R2, 5000, MPI_FLOAT, 1, 0, MPI_COMM_WORLD, &status);

    for (int i = 0; i < 5000; i++)
    {
        R[i] = R1[i];
    }
    for (int i = 5000; i < 10000; i++)
    {
        R[i] = R2[i - 5000];
    }
```

```
Windows PowerShell

Runtime = 0.000168
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Runtime = 0.000181
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Max = 3369.000000

Runtime = 0.795879
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Max = 3369.000000

Runtime = 0.796033
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Max = 3369.000000

Runtime = 0.786795
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Max = 3369.000000

Runtime = 0.802426
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug> mpie

Max = 3369.000000

Runtime = 0.483937
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\Debug>
```

```
if (rank == 1)
{
//  #pragma omp parallel
    {
        for (int i = 5000; i < 10000; i++)
        {
            while ((pow(x[0] - x[i], 2) + pow(y[0] - y[i], 2) + pow(z[0] - z[i], 2)) > R2[i - 5000] * R2[i
            {
                R2[i - 5000] = R2[i - 5000] + 1;
            }
        }
    }
    MPI_Send(&R2, 5000, MPI_FLOAT, 0, 0, MPI_COMM_WORLD);
}
else if (rank == 0) {

// #pragma omp parallel
    {
        for (int i = 5000; i < 10000; i++)
        {
            while ((pow(x[0] - x[i], 2) + pow(y[0] - y[i], 2) + pow(z[0] - z[i], 2)) > R1[i] * R1[i])
            {
                R1[i] = R1[i] + 1;
            }
        }
    }
    MPI_Status status;
    MPI_Recv(&R2, 5000, MPI_FLOAT, 1, 0, MPI_COMM_WORLD, &status);

    for (int i = 0; i < 5000; i++)
    {
        R[i] = R1[i];
    }
    for (int i = 5000; i < 10000; i++)
    {
        R[i] = R2[i - 5000];
    }
```

```
Windows PowerShell

Max = 3369.000000

Runtime = 0.795879
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D

Max = 3369.000000

Runtime = 0.796033
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D

Max = 3369.000000

Runtime = 0.786795
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D

Max = 3369.000000

Runtime = 0.802426
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D

Max = 3369.000000

Runtime = 0.483937
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D

Max = 3369.000000

Runtime = 0.793551
PS C:\Users\den19\source\repos\ConsoleApplication1\x64\D
```