



# Колебание цепочек

## Этап II

## Докладчики

Группа НФИбд-01-20:

» Михаил Ким 1032201664

- Ильин Андрей 1032201656
- Юрий Кузнецов 1032200533
- Аббас Майсаров 1032200530
- Егор Логинов 1032201661

Группа НФИбд-02-20:

- Улугбек Ибрагимов 1032204510





RUDN  
university

## Вводная часть

## Актуальность

- Колебания встречаются в большинстве природных явлений.
- С помощью колебания гармонических цепочек в частности изучаются:
  1. механические свойства материалов
  2. распространение волн
  3. исследование фазовых переходов

## Объект и предмет исследования

- Колебание цепочек
- Гармоническая цепочка
- Модель колебания гармонических цепочек

## Цель

- Построить модель гармонических цепочек.

## Задачи

- Написать программу, моделирующую поведение цепочки из  $N$  частиц. (параметры  $m = 1$ ,  $k = 1$ ,  $d = 1$ ) вместе с начальными условиями в виде гармоник с номерами  $l = \overline{1,3}$ .

## Материалы и методы

- Медведев Д. А., Куперштох А. Л., Прууэл Э. Р., Сатонкина Н. П., Карпов Д. И. Моделирование физических процессов и явлений на ПК: Учеб. пособие / Новосибирск: Новосиб. гос. ун-т., 2010. – 86-91с.
- Язык программирования Julia
- Интерактивный блокнот Pluto.jl
- Язык моделирования Modelica
- Программное обеспечение OpenModelica



## Описание модели

## Описание модели

$$x_i = i \cdot d, i = 1 \dots N$$

$$y_i = (A \cos(px_i) + B \sin(px_i)) \cos(\omega t), i = 1 \dots N$$

$$p_l = \frac{l\pi}{(N+1)d}, l = 1 \dots N, l = 1 \dots N$$

$$w_0 = \sqrt{k/m}$$

$$\omega_l = 2\omega_0 \sin\left(\frac{l\pi}{2(N+1)}\right)$$

$$y_0 = 0, y_{N+1} = 0, A = 0$$

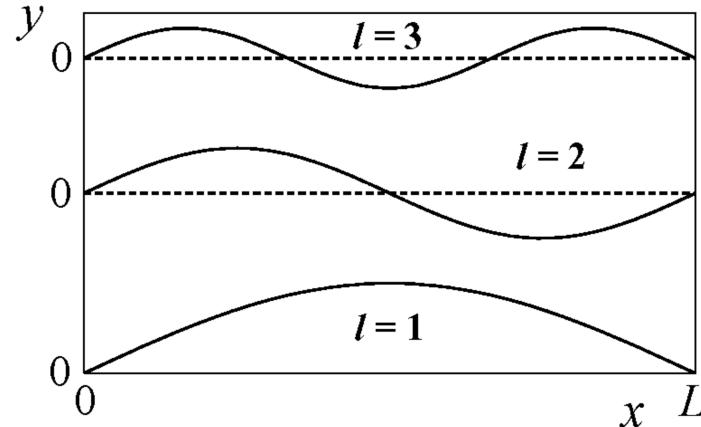


Рис. 1. Гармоники



RUDN  
university

Modelica. Построение модели

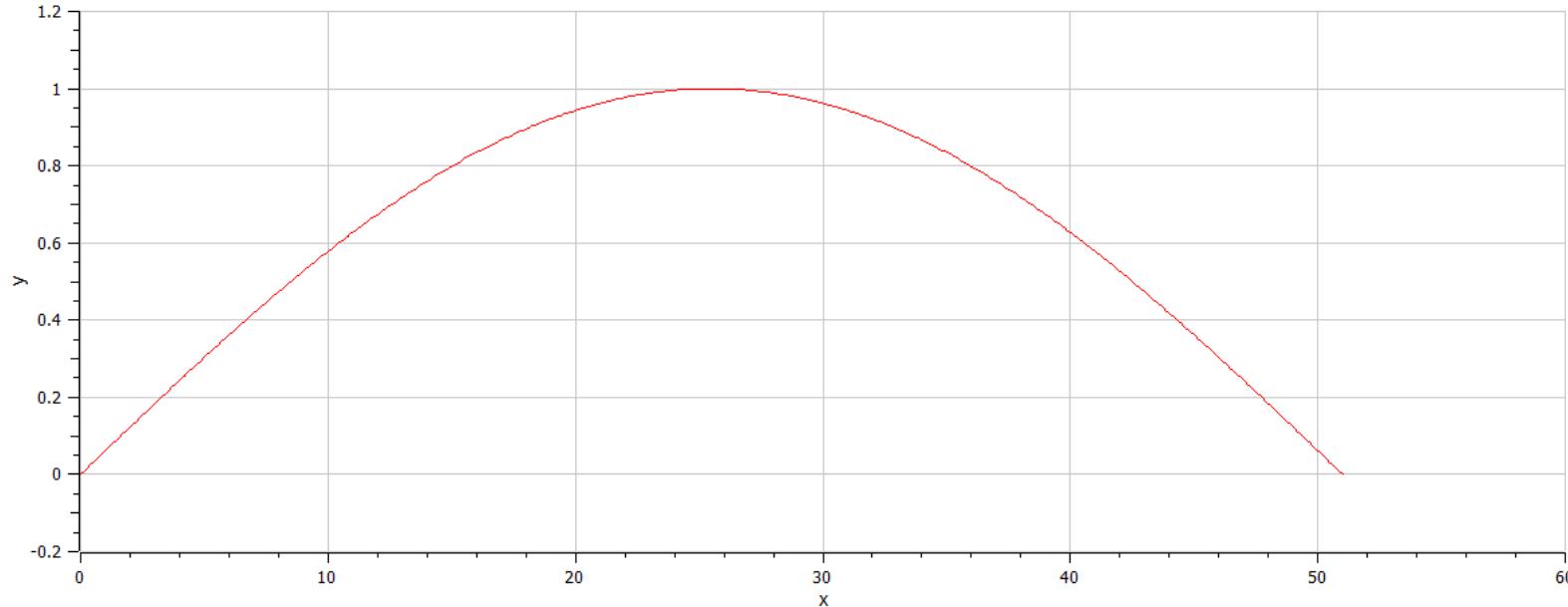


```
1 model project
2     constant Real pi = 2 * Modelica.Math.asin(1.0);
3     constant Integer N = 50;
4     constant Integer m = 1;
5     constant Integer k = 1;
6     constant Integer d = 1;
7     constant Integer A = 0;
8     constant Integer B = 1;
9     constant Real p_l = (l * pi)/((N + 1) * d);
10    constant Real omega_0 = sqrt(k/m);
11    constant Real omega_1 = 2 * omega_0 * sin((l * pi)/(2 * (N + 1)));
12    constant Integer l = 3;
13    constant Real t = 0.0;
14    Real x;
15    Real y;
16 equation
17     der(x) = 1;
18     y = (A * cos(p_l * x) + B * sin(p_l * x)) * cos(omega_1 * t);
19     annotation(experiment(StartTime=0, StopTime=51, Interval = 0.1));
20 end project;
```

## Modelica. Модель(1)

Гармоника №1

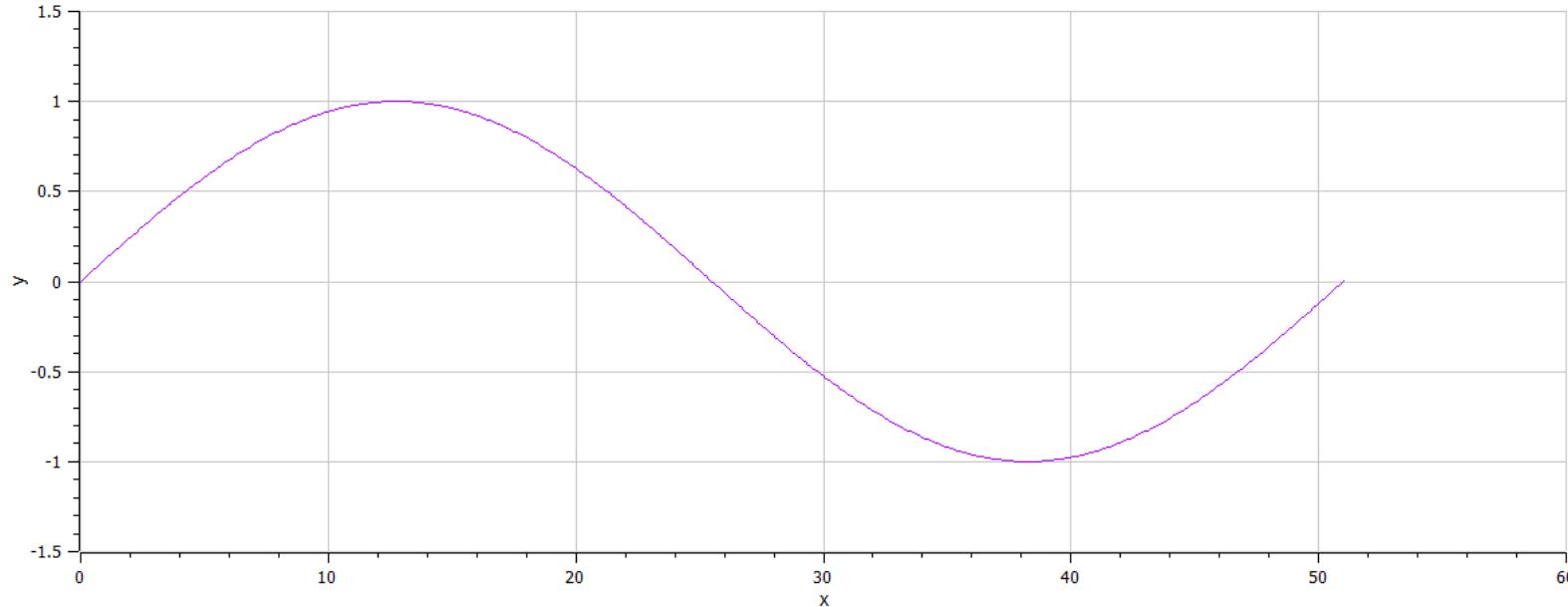
Гармоническая цепочка



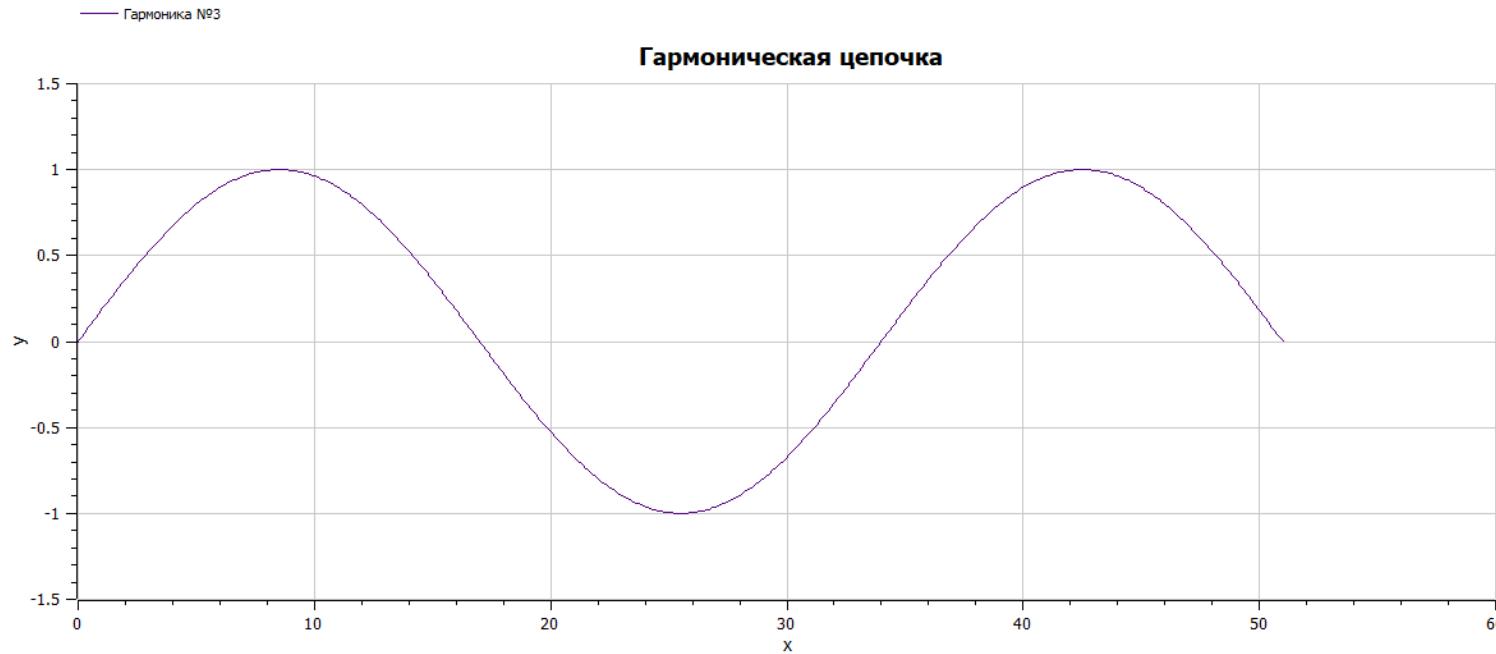
## Modelica. Модель(2)

Гармоника №2

Гармоническая цепочка



## Modelica. Модель(3)





RUDN  
university

Julia. Построение модели

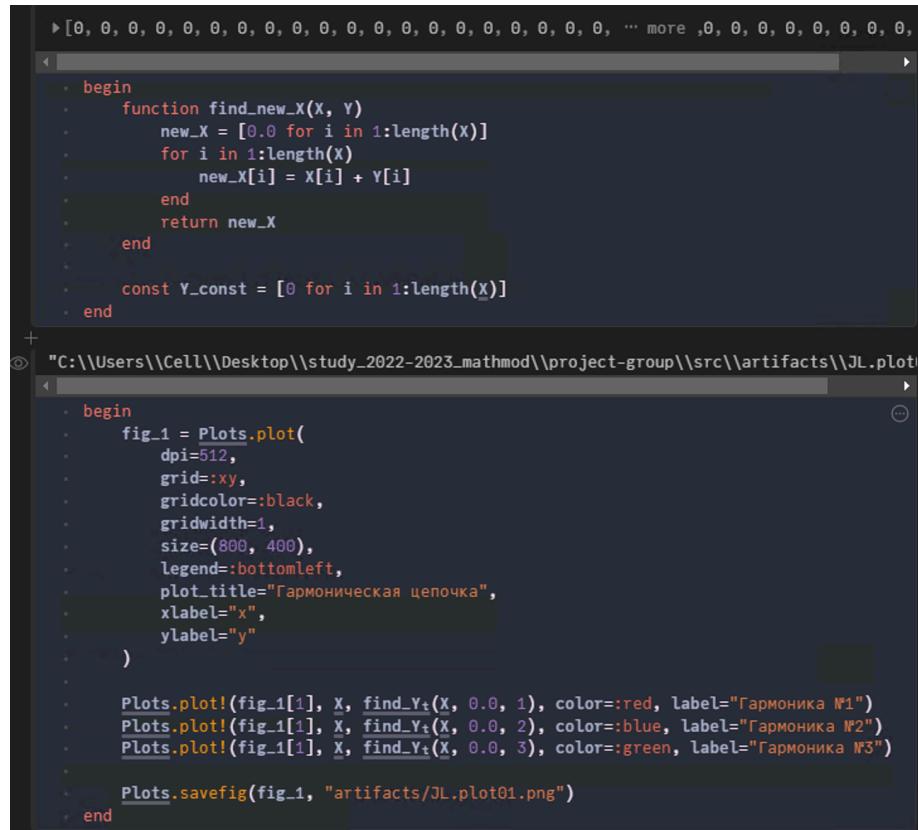
## Julia. Скрипт (1)

```
0.0:1.0:51.0
begin
    const N = 50
    const m = 1
    const k = 1
    const d = 1
    const X = [i * d for i in 0:N+1]
    const A = 0
    const B = 1
    const T = 0.0:N+1
end

find_standing_wave (generic function with 1 method)
function find_standing_wave(xi, time, l)
    pl = (l * π)/((N + 1) * d)
    ω₀ = √(k/m)
    ωl = 2 * ω₀ * sin((l * π)/(2 * (N + 1)))
    return (A * cos(pl * xi) + B * sin(pl * xi)) * cos(ωl * time)
end

find_Yt (generic function with 1 method)
function find_Yt(X, time, l)
    Yt = []
    for xi in X
        push!(Yt, find_standing_wave(xi, time, l))
    end
    return Yt
end
```

## Julia. Скрипт (2)



```
begin
    function find_new_X(X, Y)
        new_X = [0.0 for i in 1:length(X)]
        for i in 1:length(X)
            new_X[i] = X[i] + Y[i]
        end
        return new_X
    end

    const Y_const = [0 for i in 1:length(X)]
end

"C:\\\\Users\\\\Cell\\\\Desktop\\\\study_2022-2023_mathmod\\\\project-group\\\\src\\\\artifacts\\\\JL.plot"
begin
    fig_1 = Plots.plot(
        dpi=512,
        grid=:xy,
        gridcolor=:black,
        gridwidth=1,
        size=(800, 400),
        legend=:bottomleft,
        plot_title="Гармоническая цепочка",
        xlabel="x",
        ylabel="y"
    )

    Plots.plot!(fig_1[1], X, find_Y_t(X, 0.0, 1), color=:red, label="Гармоника №1")
    Plots.plot!(fig_1[1], X, find_Y_t(X, 0.0, 2), color=:blue, label="Гармоника №2")
    Plots.plot!(fig_1[1], X, find_Y_t(X, 0.0, 3), color=:green, label="Гармоника №3")

    Plots.savefig(fig_1, "artifacts/JL.plot01.png")
end
```



## Julia. Скрипт (3)

```
begin
    anim = Plots.@animate for time in 0.0:0.5:N
        fig = Plots.plot(
            layout=(3, 1),
            dpi=512,
            grid=:xy,
            gridcolor=:black,
            gridwidth=1,
            size=(800, 800),
            legend=:bottomleft,
            ylims=(-1.5, 1.5),
            plot_title="Гармоническая цепочка",
            xlabel="x",
            ylabel="y"
        )

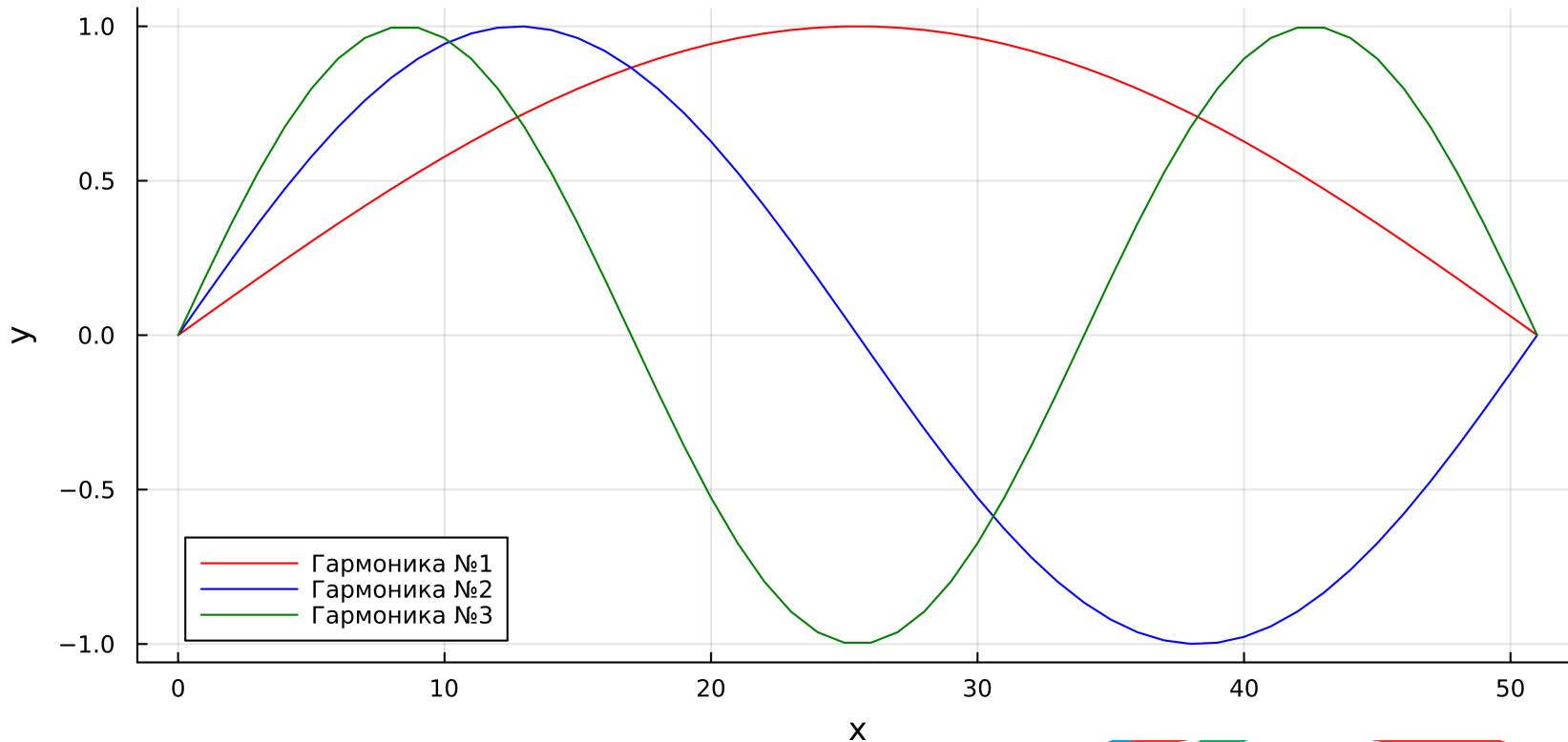
        Plots.plot!(fig[1], x, find_Y_t(x, time, 1), color=:red, label="Гармоника №1")
        Plots.scatter!(fig[1], x, Y_const, alpha=0.25, color=:grey,
        markerstrokewidth=0, label="")
        Plots.scatter!(fig[1], x, find_Y_t(x, time, 1), color=:red, label="",
        markerstrokewidth=0.4, markersize=4)

        Plots.plot!(fig[2], x, find_Y_t(x, time, 2), color=:blue, label="Гармоника №2")
        Plots.scatter!(fig[2], x, Y_const, alpha=0.25, color=:grey,
        markerstrokewidth=0, label="")
        Plots.scatter!(fig[2], x, find_Y_t(x, time, 2), color=:blue, label="",
        markerstrokewidth=0.4, markersize=4)

        Plots.plot!(fig[3], x, find_Y_t(x, time, 3), color=:green, label="Гармоника
        №3")
        Plots.scatter!(fig[3], x, Y_const, alpha=0.25, color=:grey,
        markerstrokewidth=0, label="")
        Plots.scatter!(fig[3], x, find_Y_t(x, time, 3), color=:green, label="",
        markerstrokewidth=0.4, markersize=4)
    end
    Plots.gif(anim, fps=60, "artifacts/JL.anim01.gif")
end
```

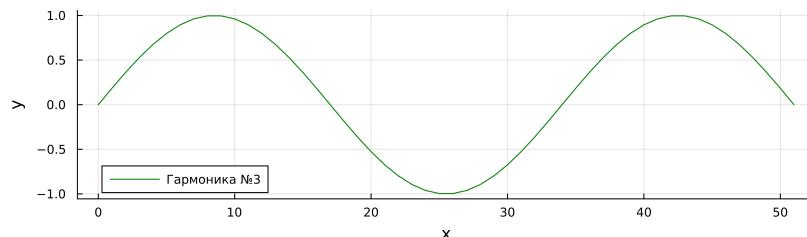
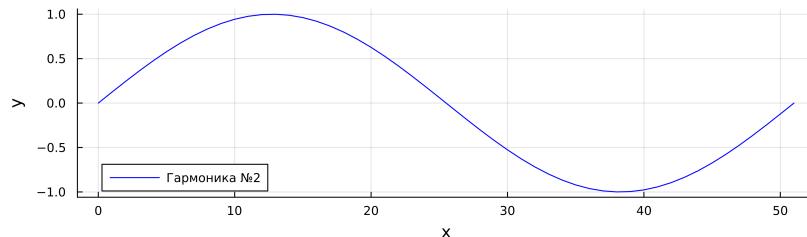
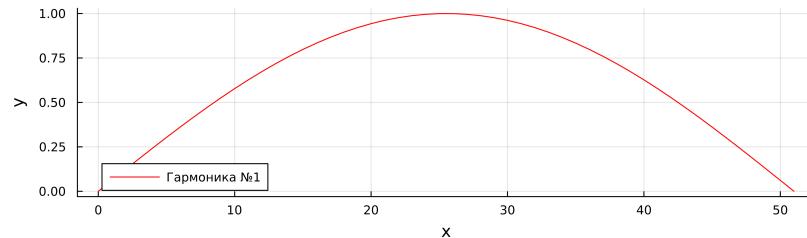
## Julia. Модель(1)

### Гармоническая цепочка

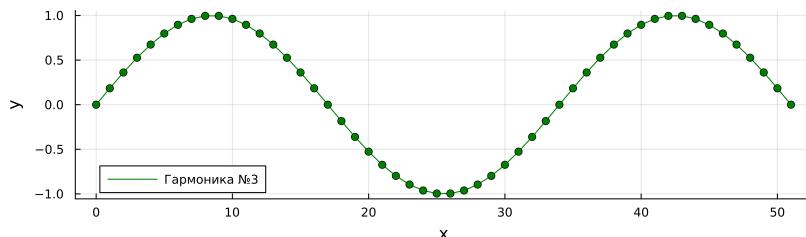
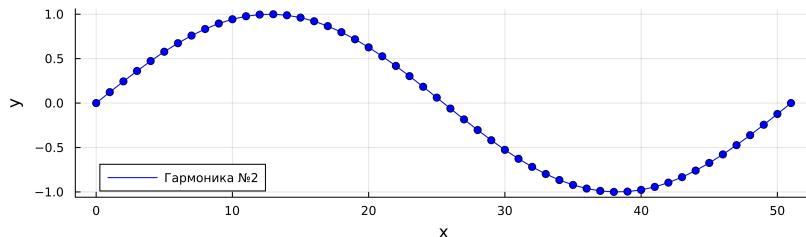
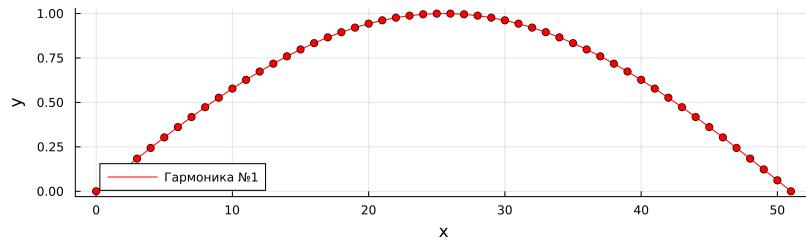


## Julia. Модель(2)

Гармоническая цепочка

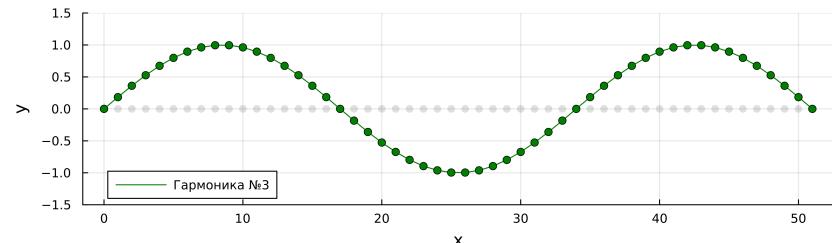
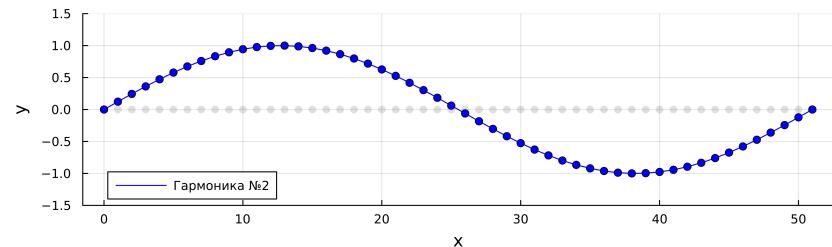
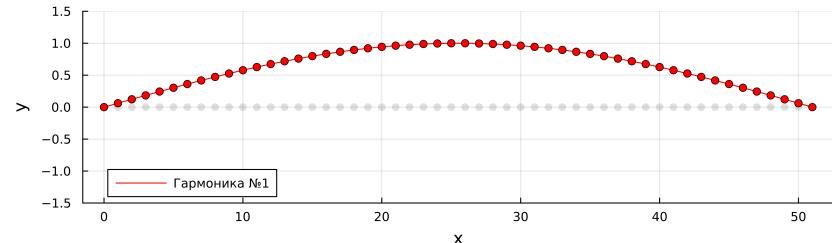


Гармоническая цепочка

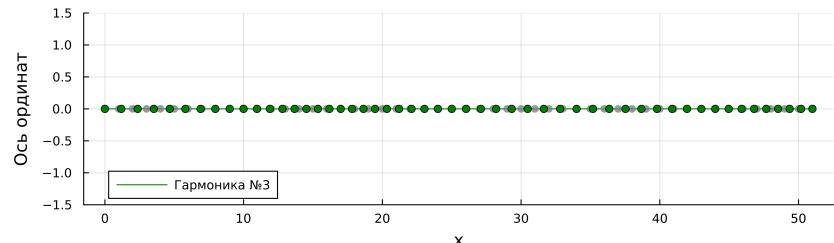
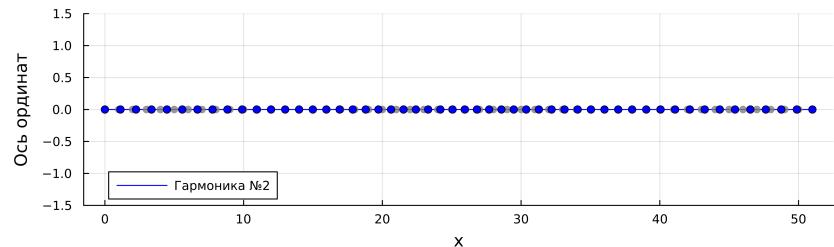
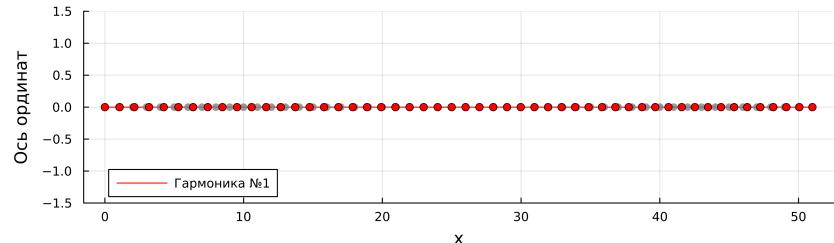


# Julia. Модель(3)

Гармоническая цепочка



Гармоническая цепочка





## Вывод

## Вывод

- Мы разработали алгоритм построения модели колебания гармонических цепочек.
- Описали особенности реализации на Julia и Modelica