Этап 2

Программная реализация решения задачи

Канева Екатерина

Клюкин Михаил

Ланцова Яна

Содержание

# 1 Цель работы

Реализовать комплекс программ для решения поставленной задачи.

# 2 Задание

1. Реализовать программу для моделирования гармонических колебаний.
2. Релализовать программу для моделирования ангармонических колебаний.

# 3 Выполнение лабораторной работы

## 3.1 Реализация алгоритма

using Plots  
using LinearAlgebra  
using FFTW  
using Dates  
  
# Гармонические колебания  
function harmonic\_chain\_simulation(;  
 N=20, # Количество частиц  
 m=1.0, # Масса частицы  
 k=1.0, # Жёсткость пружины  
 α=0.0, # Коэффициент ангармоничности (0 для гармонического случая)  
 T=100.0, # Общее время моделирования  
 Δt=0.01, # Шаг по времени  
 Δd=1.0, # Расстояние между частицами  
 initial\_displacement=0.1, # Амплитуда начального возмущения  
 save\_every=10 # Сохранять состояние каждые save\_every шагов  
)  
 # Инициализация массивов (включая граничные условия)  
 y = zeros(N+2) # Смещения (y[1] и y[N+2] - граничные условия)  
 v = zeros(N+2) # Скорости  
 a = zeros(N+2) # Ускорения  
   
 # Начальные условия - синусоидальное возмущение  
 for i in 2:N+1  
 y[i] = initial\_displacement \* sin(π\*(i-1)/N)  
 end  
   
 # Массивы для сохранения результатов  
 times = Float64[]  
 positions = Vector{Float64}[]  
 velocities = Vector{Float64}[]  
   
 # Основной цикл моделирования  
 for t in 0:Δt:T  
 # Вычисление ускорений для внутренних частиц  
 for i in 2:N+1  
 Δy\_prev = y[i] - y[i-1]  
 Δy\_next = y[i+1] - y[i]  
   
 # Гармоническая часть силы  
 F\_harmonic = k \* (y[i+1] - 2\*y[i] + y[i-1])  
   
 # Ангармоническая часть силы (если α ≠ 0)  
 F\_anharmonic = α \* (Δy\_next^3 + Δy\_prev^3)  
   
 a[i] = (F\_harmonic + F\_anharmonic) / m  
 end  
   
 # Обновление скоростей и смещений (метод Верле)  
 for i in 2:N+1  
 v[i] += a[i] \* Δt  
 y[i] += v[i] \* Δt  
 end  
   
 # Применение граничных условий  
 y[1] = 0.0  
 y[N+2] = 0.0  
   
 # Сохранение состояния (не на каждом шаге для экономии памяти)  
 if mod(round(t/Δt), save\_every) == 0  
 push!(times, t)  
 push!(positions, copy(y[2:N+1])) # Исключаем граничные точки  
 push!(velocities, copy(v[2:N+1]))  
 end  
 end  
   
 return times, positions, velocities  
end  
  
# Функция для визуализации результатов  
function plot\_chain\_dynamics(times, positions, velocities; title="")  
 # Генерируем уникальное имя файла на основе текущего времени  
 timestamp = Dates.format(now(), "yyyy-mm-dd\_HH-MM-SS")  
 filename = "chain\_dynamics\_$(timestamp).gif"  
   
 # Анимация колебаний  
 anim = @animate for (i, t) in enumerate(times)  
 p1 = plot(positions[i],   
 xlabel="Номер частицы", ylabel="Смещение",  
 title="$title, t = $(round(t, digits=2))",  
 ylims=(-maximum(abs.(positions[1]))\*1.1, maximum(abs.(positions[1]))\*1.1),  
 legend=false)  
 scatter!(p1, positions[i], color=:red)  
   
 p2 = plot(velocities[i],  
 xlabel="Номер частицы", ylabel="Скорость",  
 ylims=(-maximum(abs.(velocities[1]))\*1.1, maximum(abs.(velocities[1]))\*1.1),  
 legend=false)  
 scatter!(p2, velocities[i], color=:blue)  
   
 plot(p1, p2, layout=(2,1))  
 end  
   
 # Сохраняем анимацию в файл с уникальным именем  
 gif(anim, filename, fps=15)  
 println("Анимация сохранена в файл: ", filename)  
end  
  
# Функция для анализа спектра  
function analyze\_spectrum(positions, Δt)  
 # Анализ спектра для центральной частицы  
 central\_particle = [pos[length(pos)÷2] for pos in positions]  
 n = length(central\_particle)  
   
 # Вычисление БПФ  
 fft\_result = fft(central\_particle)  
 freqs = fftfreq(n, 1/Δt)  
 power = abs.(fft\_result).^2  
   
 # Только положительные частоты  
 idx = freqs .> 0  
 freqs = freqs[idx]  
 power = power[idx]  
   
 plot(freqs, power, xlabel="Частота", ylabel="Мощность",   
 title="Спектр колебаний центральной частицы", legend=false)  
end  
  
# Пример использования для гармонической цепочки  
times\_harmonic, positions\_harmonic, velocities\_harmonic = harmonic\_chain\_simulation(  
 N=30, T=50.0, Δt=0.05, initial\_displacement=0.2, α=0.0  
)  
  
# Визуализация  
plot\_chain\_dynamics(times\_harmonic, positions\_harmonic, velocities\_harmonic, title="Гармонические колебания")  
analyze\_spectrum(positions\_harmonic, 0.05)  
  
# Пример использования для ангармонической цепочки  
times\_anharmonic, positions\_anharmonic, velocities\_anharmonic = harmonic\_chain\_simulation(  
 N=30, T=50.0, Δt=0.05, initial\_displacement=0.2, α=0.1  
)  
  
# Визуализация  
plot\_chain\_dynamics(times\_anharmonic, positions\_anharmonic, velocities\_anharmonic, title="Ангармонические колебания")  
analyze\_spectrum(positions\_anharmonic, 0.05)

## 3.2 Полученные результаты

Выполнив моделирование, получим колебания гармонического осциллятора (рис. 1).

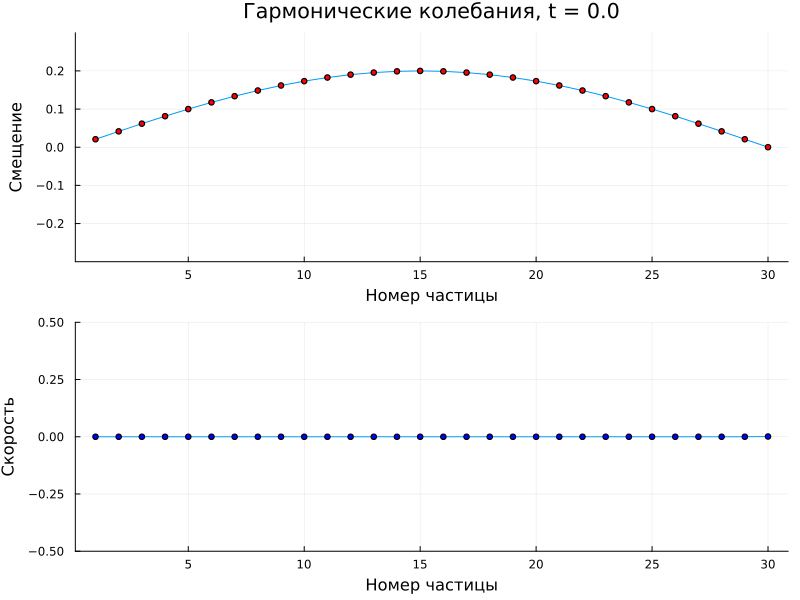


Рис. 1: Колебания гармонического осциллятора

Также смоделируем колебания ангармонического осциллятора при коэффициенте ангармоничности (рис. 2).

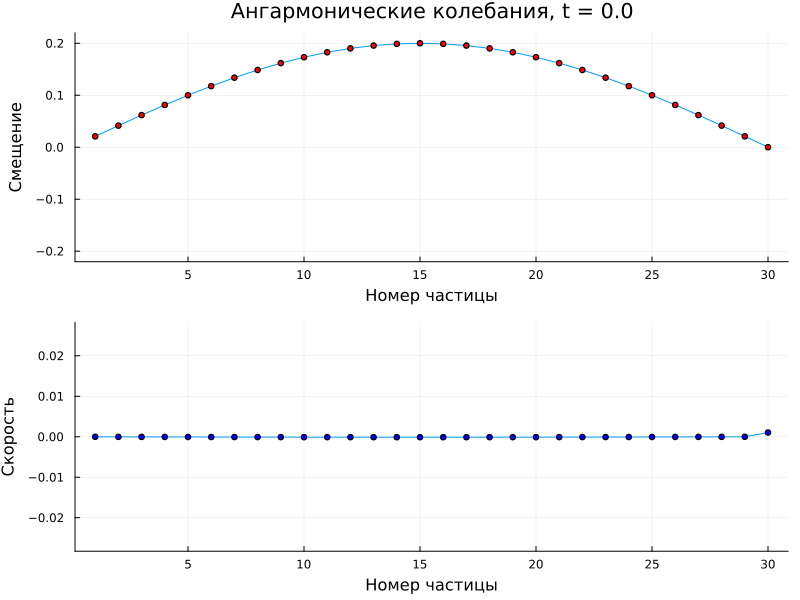


Рис. 2: Колебания ангармонического осциллятора

# 4 Выводы

Реализовать комплекс программ для решения поставленной задачи.

# Список литературы