

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

« Комп'ютерне моделювання коливань механічної системи »

(тема кваліфікаційної роботи українською мовою)

« Computer simulation of vibrations of a mechanical system »

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 126 Інформаційні системи та технології

Освітня програма Інформаційні системи та технології

Восколуп Денис Миколайович

(прізвище, ім'я, по-батькові здобувача)

Керівник д.т.н., професор Волков В.Е.

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент к.ф.-м.н., доцент Рачинська А.Л.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ ____ від ____ . ____ . 20 ____ р.

Завідувач(ка) кафедри

Алла РАЧИНСЬКА

(підпис)

Захищено на засіданні ЕК № ____

протокол № __ від ____ . ____ . 20 ____ р.

Оцінка ____ / ____ / ____

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

Володимир ВИЧУЖАНІН

(підпис)

Одеса 2023

АНОТАЦІЯ

Чисельне моделювання може бути використане для передбачення і вивчення поведінки (руху) складних фізичних і механічних систем. Для того щоб дати кількісно правильні передбачення, моделювання має адекватно описувати як окремі процеси, що діють у системі, так і їхню взаємодію.

У цій дипломній роботі виконано чисельне моделювання коливального руху для механічної системи з одним ступенем свободи. Математичне моделювання для досить простої системи з трьох фізичних тіл виявилось вельми трудомістким уже на етапі виведення рівняння її руху.

Розробка та налагодження користувацького застосунку, що безпосередньо реалізує цей чисельний експеримент, було самостійним завданням, яке вдалося успішно розв'язати в рамках наявних уявлень про подібні програми.

Застосунок було розроблено з використанням потужної мови програмування C# та надійної бібліотеки C1.Win.C1Chart.4.

ANNOTATION

Numerical modelling can be used to predict and study the behaviour (motion) of complex physical and mechanical systems. In order to give quantitatively correct predictions, simulations must adequately describe both the individual processes operating in the system and their interaction.

In this thesis we performed numerical simulation of oscillatory motion for a mechanical system with one degree of freedom. Mathematical modeling for a fairly simple system of three physical bodies turned out to be very labor-intensive already at the stage of deriving the equation of motion.

Development and debugging of the user application, which directly implements this numerical experiment, was an independent task, which was successfully solved within the limits of existing ideas about such programs.

The application was developed using the powerful C# programming language and the reliable C1.Win.C1Chart.4 library.

ЗМІСТ

1 ВСТУП	5
1.1 Визначення проблематики дослідження.....	5
1.2 Актуальність теми	6
1.3 Мета та завдання.....	7
2 МАТЕМАТИЧНА МОДЕЛЬ МЕХАНІЧНОЇ СИСТЕМИ	9
2.1 Формалізація механічної системи та її компонентів.....	9
2.2 Визначення рівнянь руху та взаємодії механічних елементів.....	10
2.3 Аналіз механічної системи	23
3 ВИБІР ПРОГРАМНОГО СЕРЕДОВИЩА ДЛЯ МОДЕЛЮВАННЯ.....	29
3.1 Критерії вибору оптимального програмного середовища.....	29
3.2 Опис обраного програмного середовища та його можливостей.....	29
4 КОМП'ЮТЕРНА МОДЕЛЬ МЕХАНІЧНОЇ СИСТЕМИ	31
4.1 Огляд інтерфейсу та функціональності застосунку.....	31
4.2 Перевірка коректності моделі	41
5 АНАЛІЗ МЕХАНІЧНОЇ СИСТЕМИ	31
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТКИ.....	47

ВСТУП

1.1 Визначення проблематики дослідження

У сучасному світі, де швидкість розвитку технологій постійно зростає, комп'ютерне моделювання стає незамінним інструментом для розуміння, проектування та вдосконалення механічних систем. Воно дозволяє інженерам, науковцям та дослідникам виконувати віртуальні експерименти, аналізувати різні варіанти проектування та прогнозувати поведінку системи без необхідності витратних фізичних експериментів.

Швидкий прогрес у сфері комп'ютерної техніки, доступність потужних обчислювальних ресурсів та розвиток операційних систем, зокрема тих, що підтримують паралельні обчислення, значно збільшують зацікавленість у проектуванні та реалізації чисельних експериментів в фізиці та механіці за допомогою інформаційних систем.

Комп'ютерне моделювання коливань механічних систем є важливою галуззю досліджень, яка знаходить широке застосування у багатьох сферах, включаючи інженерію, науку, промисловість та інші галузі. Застосування комп'ютерного моделювання коливань механічних систем дозволяє отримати глибоке розуміння їх поведінки та властивостей, а також забезпечує можливість вдосконалення та оптимізації цих систем.

У сучасному світі інженери та дослідники стикаються з різноманітними завданнями, пов'язаними з механічними системами, які піддаються коливанням. Наприклад, в автомобільній промисловості комп'ютерне моделювання коливань може бути використане для аналізу динамічної стійкості автомобілів, визначення причин вібрацій або шумів, а також для проектування оптимальних систем підвіски. У виробництві машин комп'ютерне моделювання коливань може допомогти в розробці ефективних амортизаторів або уникненні резонансних явищ.

В галузі будівництва комп'ютерне моделювання коливань механічних систем може бути використане для аналізу поведінки будівель, мостів або

споруд при впливі землетрусів або вітрових навантажень. Це дозволяє забезпечити безпеку та стійкість будівельних конструкцій.

У науці комп'ютерне моделювання коливань механічних систем є невід'ємною частиною вивчення різних фізичних явищ. Воно дозволяє розуміти складну динаміку систем, проводити чисельні експерименти та дослідження, які не завжди можуть бути виконані на практиці.

Крім того, комп'ютерне моделювання коливань механічних систем має важливість і в інших сферах, таких як аерокосмічна промисловість, медицина, спорт та біомеханіка. Всюди, де виникають коливання механічних систем, моделювання може допомогти визначити їх характеристики, вплив зовнішніх факторів та розробити методи їх контролю.

Таким чином, комп'ютерне моделювання коливань механічних систем має велике значення в різних галузях, допомагаючи покращувати розуміння та оптимізацію механічних систем, забезпечуючи безпеку, ефективність та стійкість їх функціонування.

1.2 Актуальність теми

Існують деякі актуальні проблеми, пов'язані з коливаннями механічних систем і які можуть бути вирішені за допомогою комп'ютерного моделювання. Коливання механічних систем є складною проблемою, яка стикається з різними викликами і потребує вдосконалення методів аналізу та проектування.

Однією з основних проблем є визначення динамічної стійкості механічних систем. При високих швидкостях або в умовах змінних навантажень системи можуть втратити стійкість і стати причиною аварій. Комп'ютерне моделювання дозволяє вивчати вплив різних факторів на стійкість системи, виявляти критичні точки і розробляти методи їх попередження.

Для досягнення кращої ефективності та зниження ваги механічних систем необхідна оптимізація їх конструкції. Комп'ютерне моделювання

дозволяє проводити чисельні експерименти з різними варіантами конструкцій та матеріалів, визначати їх вплив на коливання системи і знаходити оптимальні рішення.

Колівання механічних систем можуть бути викликані як зовнішніми навантаженнями (наприклад, вібрація або удари), так і внутрішніми факторами (наприклад, неоднорідність матеріалу або не ідеальність з'єднань). Комп'ютерне моделювання дозволяє вивчати вплив цих факторів на коливання системи, ідентифікувати критичні точки та розробляти заходи для зниження їх впливу.

З розвитком технологій з'являються все складніші механічні системи, такі як роботи, автоматизовані виробничі лінії або транспортні системи. Аналіз та проектування таких систем вимагає нових підходів і методів. Комп'ютерне моделювання дозволяє вивчати взаємодію складних компонентів системи, аналізувати їх динаміку та ефективність, ідентифікувати проблемні зони та знаходити шляхи їх вирішення.

Всі ці проблеми підкреслюють необхідність розвитку нових підходів до аналізу та проектування механічних систем. Комп'ютерне моделювання є потужним інструментом, який дозволяє вирішувати ці проблеми шляхом створення точних і реалістичних моделей, проведення чисельних експериментів та визначення оптимальних рішень.

Таким чином, комп'ютерне моделювання дозволяє вирішувати ряд актуальних проблем, пов'язаних з коливаннями механічних систем, та покращувати процес аналізу та проектування з точки зору інформаційних систем. Розвиток нових підходів до комп'ютерного моделювання може допомогти впровадженню більш точних, швидких та ефективних методів розрахунків та оптимізації механічних систем.

1.3 Мета та завдання

Метою цієї роботи є чисельне моделювання коливального руху для механічної системи з одним ступенем свободи. А також розробка та налагодження користувацького застосунку, що безпосередньо реалізує цей чисельний експеримент.

Задана механічна система з одним ступенем свободи може чинити коливання відносно положення рівноваги. В початковий момент часу $t = 0$ систему виводять із положення рівноваги шляхом вертикального усунення вантажу 1, яке не порушує цілісності самої системи. Швидкості всіх точок системи при $t = 0$ дорівнюють нулю (система в стані спокою).

У наступний момент часу система починає рух (коливання), перебуваючи під впливом лише консервативних сил.

Для досягнення цієї мети необхідно :

1. Скласти диференційне рівняння, що описує рух заданої механічної системи (вільні коливання).
2. З'ясувати умови, за яких дана механічна система здійснюватиме коливання біля положення рівноваги.
3. Створити додаток користувача, що реалізує чисельний експеримент з даної задачі і виконує візуалізацію основних рухів.
4. За допомогою чисельного інтегрування (методом Рунге-Кутта) визначити рішення диференціального рівняння руху за заданих початкових умов.
5. За результатами чисельного інтегрування визначити циклічну частоту k та період τ вільних коливань заданої механічної системи
6. Порівняти параметри цього процесу з випадком мінімальних коливань. прогнозування її поведінки

МАТЕМАТИЧНА МОДЕЛЬ МЕХАНІЧНОЇ СИСТЕМИ

2.1 Формалізація механічної системи та її компонентів

Схема механічної системи наведена у стані спокою, див. Рисунок 1.

Як узагальнена координата ξ приймається вертикальне становище вантажу 1.

Нижче наведено фізичні характеристики даної механічної системи, які вважаються заданими (або задаються під час чисельного експерименту).

Маси тіл системи – m_1 , m_2 и m_3 ; c – коефіцієнт відновлення для пружини.

Розміри:

R – зовнішній радіус тіла 2;

r – внутрішній радіус тіла 2, при цьому $r < R$;

L – довжина тіла 3 (однорідний стрижень), при цьому $R \leq L$;

L_c – дистанція від точки підвісу тіла 3 до точки закріплення пружини;

Δ_0 – довжина недеформованої пружини, розміщеної горизонтально;

Δ_{cm} – деформація пружини у стані спокою системи;

H – довжина зв'язку (нерозтяжний стрижень)

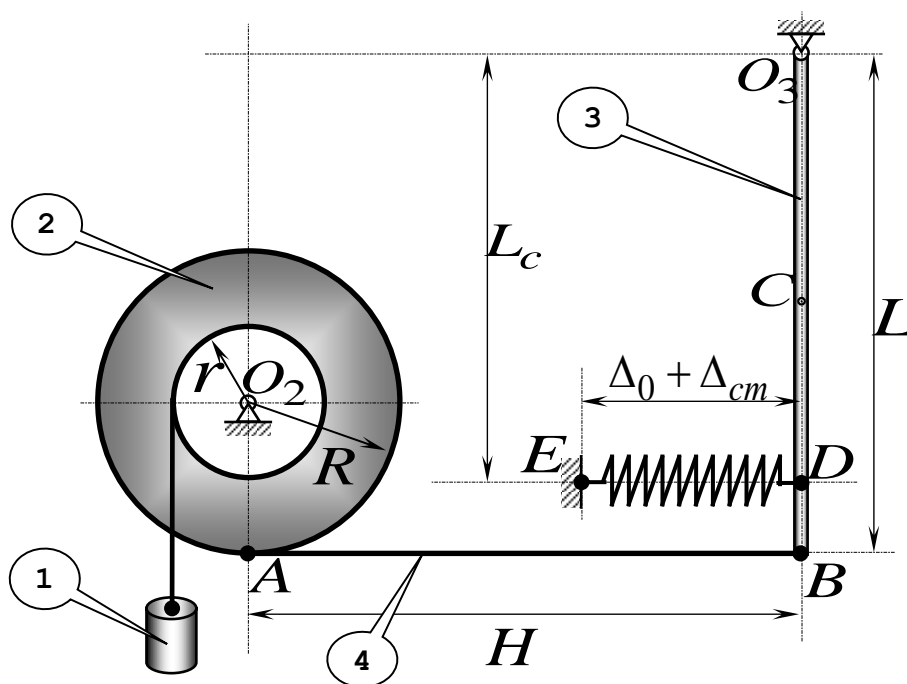


Рисунок 1. Задана механічна система у стані рівноваги (спокою).

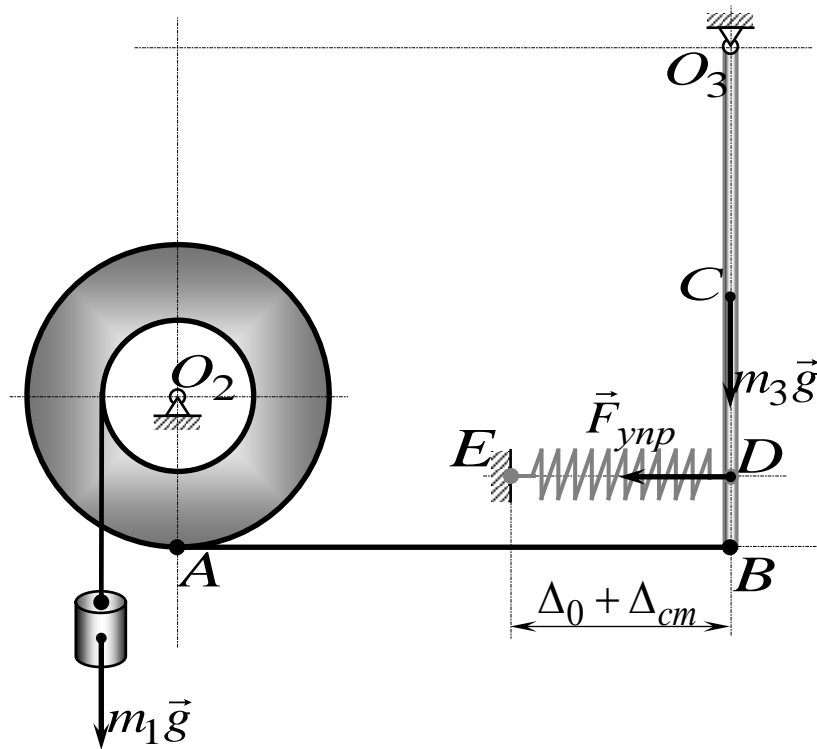


Рисунок 2. Сукупність зовнішніх сил, що забезпечує стан спокою.

2.2 Визначення рівнянь руху та взаємодії механічних елементів

Розглянута механічна система, що має один ступінь свободи, складається з трьох фізичних тіл, геометричних зв'язків та пружини.

Для дослідження динаміки руху цієї системи будемо застосовувати рівняння Лагранжа 2-го роду:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} = - \frac{\partial \Pi}{\partial \xi} + Q_{\xi},$$

де ξ і $\dot{\xi}$ – відповідно узагальнена координата, та узагальнена швидкість; T – Кінетична енергія механічної системи; Π – її потенційна енергія; Q_{ξ} – узагальнена сила (при наявності непотенційних сил).

В якості узагальної координати ξ виберемо вертикальне відхилення вантажу 1 від положення рівноваги у напрямку дії сили тяжіння $m_1 \vec{g}$.

Відповідно узагальнену швидкість вантажу 1 позначимо $\dot{\xi}$. Кінетична енергія даного тіла при його вертикальному поступальному русі буде:

$$T_1 = \frac{1}{2} m_1 \dot{\xi}^2.$$

Виразимо кінематичні характеристики решти тіл через узагальнену швидкість $\dot{\xi}$ вантажу 1, це можливо, оскільки задана система має один ступінь свободи.

Тіло 2 здійснює обертальний рух навколо горизонтальної осі, що проходить через геометричний центр O_2 .

Положення цього тіла будемо визначати за допомогою кута повороту ϕ навколо цієї осі. Тоді кутова швидкість $\dot{\phi}$ тіла 2 може бути обчислена за формулою Ейлера: $\dot{\phi} = \dot{\xi}/r$.

Кінетична енергія тіла 2 обчислюється за формулою $T_2 = \frac{1}{2} J_2 \dot{\phi}^2$, де J_2 – момент інерції тіла 2 відносно горизонтальної осі обертання, що проходить через центр O_2 .

Швидкість точки A , що лежить на зовнішньому радіусі R тіла 2, матиме величину $V_A = R \cdot \dot{\phi} = (R/r) \cdot \dot{\xi}$.

Тіло 3 (однорідний стрижень довжини L) також здійснює обертальний рух. Горизонтальна вісь цього обертання проходить через точку підвісу O_3 . Кут відхилення тіла 3 від вертикалі позначимо через ψ . Швидкість точки B стрижня 3 може бути обчислена за формулою $V_B = L \cdot \dot{\psi}$, де $\dot{\psi}$ – кутова швидкість цього тіла.

Кінетична енергія тіла 3 обчислюється за формулою $T_3 = \frac{1}{2} J_3 \dot{\psi}^2$, де $J_3 = m_3 L^2 / 3$ – момент інерції відносно горизонтальної осі обертання, що проходить через центр O_3 .

Як уже зазначалося вище, механічна система, що розглядається, має один ступінь свободи, тому кутові швидкості $\dot{\varphi}$ та $\dot{\psi}$ тіл 2 і 3 повинні однозначно функціонально виражатися через узагальнену швидкість $\dot{\xi}$.

Тіла 2 і 3 з'єднуються за допомогою невагомий стрижня AB довжини H . В точках A і B цей стрижень має специфічні шарніри. Таким чином, даний стрижень є ідеальним недеформованим зв'язком.

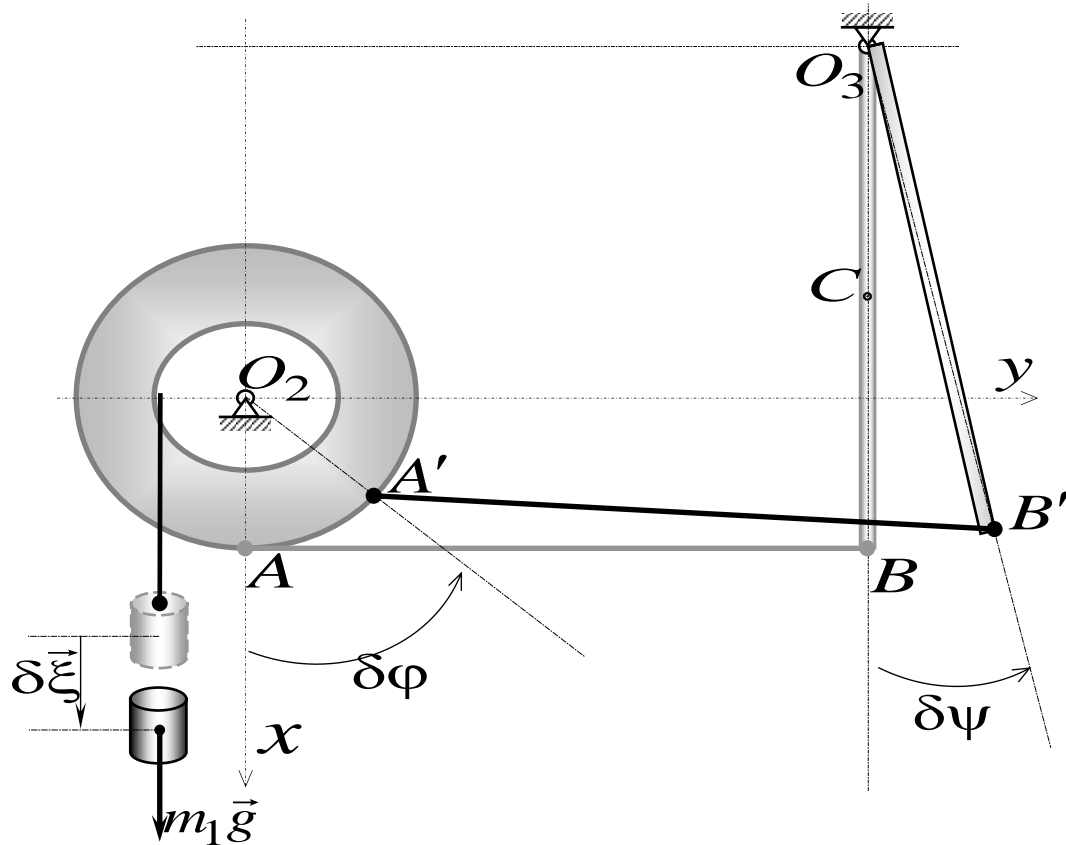


Рисунок 3. Віртуальні рухи тіл механічної системи.

Задамо віртуальне переміщення $\delta \vec{\xi}$ вантажу 1 у напрямі дії сили тяжіння.

Відповідно до цього переміщення тіла 2 і 3 також отримають переміщення, що представляють собою повороти на певні кути $\delta \varphi$ і $\delta \psi$, см.

Рисунок 3.

При цьому точки A і B тіл 2 і 3 займуть відповідні положення A' і B' .

Відстань між точками A' і B' через жорсткість зв'язку AB залишиться такою ж, тобто $|AB| = |A'B'| = H$.

Зв'яжемо з центром O_2 систему координат O_2xy , см. Рисунок 3.

Центр O_3 у цих осях матиме координати: $x_{O_3} = R - L$ і $y_{O_3} = H$, (що можна побачити з рисунку 1).

Координати точки A будуть: $x_A = R \cdot \cos \varphi$ і $y_A = R \cdot \sin \varphi$.

Координати точки B будуть: $x_B = R - L + L \cdot \cos \psi$ і $y_B = H + L \cdot \sin \psi$.

Тоді незмінність довжини H стрижня AB під час руху системи може бути виражена співвідношенням $(x_B - x_A)^2 + (y_B - y_A)^2 = H^2$.

Цьому співвідношенню надамо форму рівняння зв'язку:

$$f(\varphi, \psi) = (x_B - x_A)^2 + (y_B - y_A)^2 - H^2 = 0, \quad \text{або}$$

$$f(\varphi, \psi) = (R \cdot (1 - \cos \varphi) - L \cdot (1 - \cos \psi))^2 + (H + L \cdot \sin \psi - R \cdot \sin \varphi)^2 - H^2 = 0. \quad (1)$$

Тут R , L і H грають роль незмінних параметрів (заданих величин).

Фактично рівняння (1) пов'язує неявним чином змінні ψ і ξ з врахуванням того, що $\dot{\varphi} = \dot{\xi}/r$:

$$f(\xi, \psi) = (R \cdot (1 - \cos(\xi/r)) - L \cdot (1 - \cos \psi))^2 + (H + L \cdot \sin \psi - R \cdot \sin(\xi/r))^2 - H^2 = 0. \quad (2)$$

За наявності зв'язку, що описується рівнянням (1), варіації $\delta\varphi$ і $\delta\psi$ кутових координат φ і ψ повинні задовільнити співвідношенню

$$\frac{\partial f}{\partial \varphi} \delta\varphi + \frac{\partial f}{\partial \psi} \delta\psi = 0. \quad (3)$$

Знайдемо часткові похідні, що входять до останнього співвідношення:

$$\begin{aligned} \frac{\partial f}{\partial \varphi} &= 2(R \cdot (1 - \cos \varphi) - L \cdot (1 - \cos \psi)) \cdot R \cdot \sin \varphi - 2(H + L \cdot \sin \psi - R \cdot \sin \varphi) \cdot R \cdot \cos \varphi, \\ \frac{\partial f}{\partial \psi} &= -2(R \cdot (1 - \cos \varphi) - L \cdot (1 - \cos \psi)) \cdot L \cdot \sin \psi + 2(H + L \cdot \sin \psi - R \cdot \sin \varphi) \cdot L \cdot \cos \psi. \end{aligned} \quad (4)$$

Тепер ми можемо переписати співвідношення (3) для варіацій $\delta\varphi$ і $\delta\psi$ у вигляді:

$$\mathbf{X} \cdot \delta\varphi - \mathbf{Z} \cdot \delta\psi = 0 \quad \text{чи} \quad \delta\psi = \frac{\mathbf{X}}{\mathbf{Z}} \cdot \delta\varphi, \quad (5)$$

$$\begin{aligned} \mathbf{X}(\varphi, \psi) &= 2R \cdot \left[(R \cdot (1 - \cos \varphi) - L \cdot (1 - \cos \psi)) \cdot \sin \varphi - (H + L \cdot \sin \psi - R \cdot \sin \varphi) \cdot \cos \varphi \right], \\ \text{де } \mathbf{Z}(\varphi, \psi) &= 2L \cdot \left[(R \cdot (1 - \cos \varphi) - L \cdot (1 - \cos \psi)) \cdot \sin \psi - (H + L \cdot \sin \psi - R \cdot \sin \varphi) \cdot \cos \psi \right]. \end{aligned} \quad (6)$$

Співвідношення (6) буде справедливим і для кутових швидкостей тіл 2 і 3, тобто

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{d\varphi}{dt} = \frac{\mathbf{X}}{\mathbf{Z}} \cdot \dot{\varphi}. \quad (7)$$

Отже, нам вдалося виразити кутові швидкості $\dot{\varphi}$ і $\dot{\psi}$ тіл 2 і 3 через узагальнену швидкість $\dot{\xi}$:

$$\dot{\varphi} = \frac{1}{r} \cdot \dot{\xi}, \quad \dot{\psi} = \frac{1}{r} \cdot \frac{\mathbf{X}}{\mathbf{Z}} \cdot \dot{\xi}. \quad (8)$$

Повна кінетична енергія розглянутої механічної системи має такий вигляд:

$$T = T_1 + T_2 + T_3 = \frac{1}{2} \left(m_1 \dot{\xi}^2 + J_2 \dot{\varphi}^2 + J_3 \dot{\psi}^2 \right).$$

Підстановка (8) в останню формулу дає:

$$T = \frac{1}{2r^2} \left(m_1 r^2 + J_2 + J_3 \cdot \Phi(\varphi, \psi) \right) \cdot \dot{\xi}^2, \quad (9)$$

$$\text{де } \Phi = \Phi(\varphi, \psi) = \left[\frac{\mathbf{X}}{\mathbf{Z}} \right]^2. \quad (10)$$

Моменти інерції, що входять до формули (9), можна записати таким чином:

$$J_1 = m_1 r^2, \quad J_2 = \frac{m_2}{2} \frac{r^4 + R^4}{r^2 + R^2} \quad \text{и} \quad J_3 = \frac{m_3 L^2}{3}, \quad (11)$$

де m_2 и m_3 – відповідно маси тіл 2 і 3.

Тоді повна кінетична енергія системи буде:

$$T = \frac{1}{2r^2} \left(J_1 + J_2 + J_3 \cdot \Phi \right) \cdot \dot{\xi}^2. \quad (12)$$

Для випадку малих коливань, коли відрізок AB здійснює рух близький до поступального (тобто $V_A \square V_B$), основні розрахункові формули істотно спрощуються:

$$\delta\psi = \frac{1}{r} \frac{R}{L} \cdot \delta\xi, \quad \dot{\psi} = \frac{1}{r} \frac{R}{L} \cdot \dot{\xi}, \quad \frac{\mathbf{X}}{\mathbf{Z}} = \frac{R}{L}, \quad \Phi_*(\varphi, \psi) = \left(\frac{R}{L}\right)^2 \text{ і } T_* = \frac{1}{2r^2} \left(J_1 + J_2 + J_3 \frac{R^2}{L^2} \right) \cdot \dot{\xi}^2. \quad (13)$$

В рівняння Лагранжа 2-го роду

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} = - \frac{\partial \Pi}{\partial \xi} + Q_\xi \quad (14)$$

входять часткові похідні кінетичної енергії та потенційної енергії за узагальненою координатою ξ , приватна похідна кінетичної енергії за узагальненою швидкістю $\dot{\xi}$ і узагальнена сила Q_ξ (необхідна для непотенційних сил).

Безпосередньо з (12) маємо

$$\frac{\partial T}{\partial \xi} = \frac{J_3}{2r^2} \cdot \dot{\xi}^2 \cdot \frac{\partial \Phi}{\partial \xi}, \quad \frac{\partial T}{\partial \dot{\xi}} = \frac{1}{r^2} (J_1 + J_2 + J_3 \cdot \Phi) \cdot \dot{\xi}. \quad (15)$$

Вираз для похідної $\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right)$ представимо в такому вигляді:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) = \frac{d}{dt} \left(\frac{1}{r^2} (J_1 + J_2 + J_3 \cdot \Phi) \cdot \dot{\xi} \right) = \frac{J_1 + J_2 + J_3 \cdot \Phi}{r^2} \cdot \ddot{\xi} + \frac{J_3}{r^2} \frac{d\Phi}{dt} \cdot \dot{\xi}. \quad (16)$$

Для обчислення другого доданка скористаємося поданням $\frac{d\Phi}{dt} = \frac{\partial \Phi}{\partial \xi} \dot{\xi} + \frac{\partial \Phi}{\partial t}$. Оскільки в нашому випадку $\frac{\partial \Phi}{\partial t} = 0$, то $\frac{d\Phi}{dt} = \frac{\partial \Phi}{\partial \xi} \cdot \dot{\xi}$ і формула (16) переписується в вигляді:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) = \frac{J_1 + J_2 + J_3 \cdot \Phi}{r^2} \cdot \ddot{\xi} + \frac{J_3}{r^2} \frac{\partial \Phi}{\partial \xi} \cdot \dot{\xi}^2.$$

Ліва частина рівняння Лагранжа матиме вигляд:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} &= \frac{J_1 + J_2 + J_3 \cdot \Phi}{r^2} \cdot \ddot{\xi} + \frac{J_3}{r^2} \frac{\partial \Phi}{\partial \xi} \cdot \dot{\xi}^2 - \frac{J_3}{2r^2} \frac{\partial \Phi}{\partial \xi} \cdot \dot{\xi}^2 = \\ &= \frac{J_1 + J_2 + J_3 \cdot \Phi}{r^2} \cdot \ddot{\xi} + \frac{J_3}{2r^2} \frac{\partial \Phi}{\partial \xi} \cdot \dot{\xi}^2. \end{aligned}$$

Очевидно (дивитися (10)), що
$$\frac{\partial \Phi}{\partial \xi} = 2 \left[\frac{\mathbf{X}}{\mathbf{Z}} \right] \cdot \frac{\partial [\mathbf{X}/\mathbf{Z}]}{\partial \xi} = 2 \frac{\mathbf{X}}{\mathbf{Z}^3} \cdot \left(\frac{\partial \mathbf{X}}{\partial \xi} \cdot \mathbf{Z} - \mathbf{X} \cdot \frac{\partial \mathbf{Z}}{\partial \xi} \right).$$

Далі
$$\frac{\partial \mathbf{X}}{\partial \xi} = \frac{\partial \mathbf{X}}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial \xi} + \frac{\partial \mathbf{X}}{\partial \psi} \cdot \frac{\partial \psi}{\partial \xi} \quad \text{и} \quad \frac{\partial \mathbf{Z}}{\partial \xi} = \frac{\partial \mathbf{Z}}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial \xi} + \frac{\partial \mathbf{Z}}{\partial \psi} \cdot \frac{\partial \psi}{\partial \xi}, \quad (17)$$

де відповідно до с (8):
$$\frac{\partial \varphi}{\partial \xi} = \frac{1}{r} \quad \text{и} \quad \frac{\partial \psi}{\partial \xi} = \frac{1}{r} \cdot \frac{\mathbf{X}}{\mathbf{Z}}, \quad (18)$$

і
$$\begin{cases} \frac{\partial \mathbf{X}}{\partial \varphi} = 2R \cdot [(R-L) \cdot \cos \varphi + H \cdot \sin \varphi + L \cdot \cos(\varphi - \psi)], \\ \frac{\partial \mathbf{X}}{\partial \psi} = 2R \cdot [-L \cdot \sin \psi \sin \varphi - L \cdot \cos \psi \cos \varphi] = -2RL \cdot \cos(\varphi - \psi), \\ \frac{\partial \mathbf{Z}}{\partial \varphi} = 2L \cdot [R \cdot \sin \psi \sin \varphi + R \cdot \cos \psi \cos \varphi] = 2RL \cdot \cos(\varphi - \psi), \\ \frac{\partial \mathbf{Z}}{\partial \psi} = 2L \cdot [(R-L) \cdot \cos \psi + H \cdot \sin \psi - R \cos(\varphi - \psi)]. \end{cases} \quad (19)$$

З урахуванням виразів (17)-(19) формула для обчислення $\frac{\partial \Phi}{\partial \xi}$ буде мати вигляд:

$$\frac{\partial \Phi}{\partial \xi} = 2 \left[\frac{\mathbf{X}}{\mathbf{Z}} \right] \cdot \frac{\partial [\mathbf{X}/\mathbf{Z}]}{\partial \xi} = \frac{2}{r} \frac{\mathbf{X}}{\mathbf{Z}^3} \cdot \left[\left(\frac{\partial \mathbf{X}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{X}}{\partial \psi} \right) \cdot \mathbf{Z} - \mathbf{X} \cdot \left(\frac{\partial \mathbf{Z}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{Z}}{\partial \psi} \right) \right]. \quad (20)$$

Остаточно отримаємо:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \Gamma}{\partial \dot{\xi}} \right) - \frac{\partial \Gamma}{\partial \xi} &= \frac{J_1 + J_2 + J_3 \cdot [\mathbf{X}/\mathbf{Z}]^2}{r^2} \cdot \ddot{\xi} + \\ &+ \frac{J_3}{r^3} \frac{\mathbf{X}}{\mathbf{Z}^3} \cdot \left[\left(\frac{\partial \mathbf{X}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{X}}{\partial \psi} \right) \cdot \mathbf{Z} - \mathbf{X} \cdot \left(\frac{\partial \mathbf{Z}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{Z}}{\partial \psi} \right) \right] \cdot \dot{\xi}^2. \end{aligned} \quad (21)$$

Перейдемо до розгляду потенційної енергії заданої механічної системи.

На фізичні тіла досліджуваної системи діють тільки сили тяжіння та пружна сила пружини.

Зв'язки, накладені на систему, вважаються ідеальними, а силами опору нехтуємо за умовою задачі.

Тому ми можемо стверджувати, що ця механічна система є консервативною - її повна механічна енергія зберігатиметься під час руху.

Отже, якщо в початковий момент часу механічна система буде відхилена від положення рівноваги та відпущена без початкової швидкості, то її подальший рух буде коливанням біля цього положення рівноваги.

Потенційна енергія системи є результат дії на систему зовнішніх сил тяжіння та пружності.

Вантаж 1 здійснює вертикальний поступальний рух.

Його потенційна енергія дорівнюватиме роботі сили тяжіння на переміщення тіла 1 з даного положення (що відповідає узагальненій координаті ξ , см. Рис. 3) у положення його рівноваги:

$$\Pi_1 = -m_1 g \cdot \xi, \quad \frac{\partial \Pi_1}{\partial \xi} = -m_1 g. \quad (22)$$

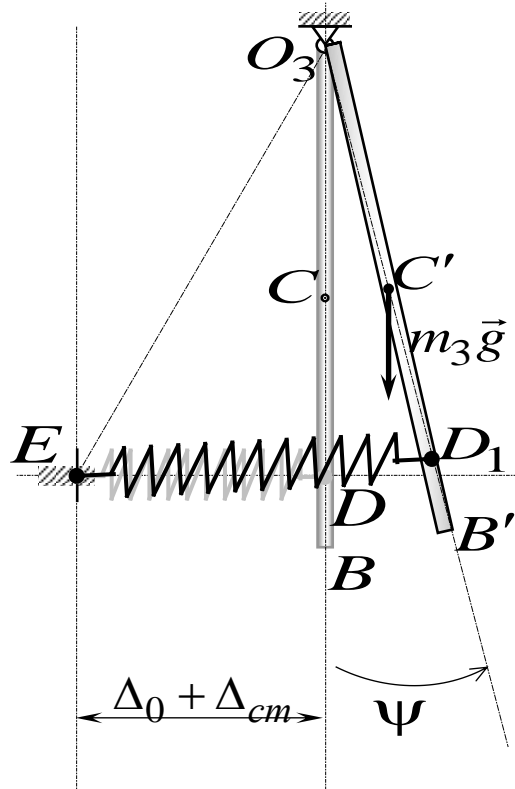


Рисунок 4.

Потенційна енергія, що відповідає силі тяжіння $m_3 \vec{g}$, прикладеної до тіла 3, дорівнюватиме роботі цієї сили під час переміщення його центру ваги з положення C' в положення C , см. Рисунок 4:

$$\Pi_3 = \frac{L}{2}(1 - \cos \psi) \cdot m_3 g, \quad \frac{\partial \Pi_3}{\partial \xi} = \frac{L}{2} m_3 g \cdot \sin \psi \cdot \frac{\partial \psi}{\partial \xi}, \quad (23)$$

де значення ψ визначається з рівняння (2), а величина $\frac{\partial \psi}{\partial \xi}$ – з виразу (18).

Знайдемо тепер вираз для узагальненої сили $Q_{\text{упр}}$, що відповідає пружній силі пружини.

За визначенням $\delta A = -M_{\text{упр}} \cdot \delta \psi = Q_{\text{упр}} \cdot \delta \xi$, где $M_{\text{упр}}$ – момент пружної сили відносно центру O_3 .

Розглянемо схему, зображену нижче:

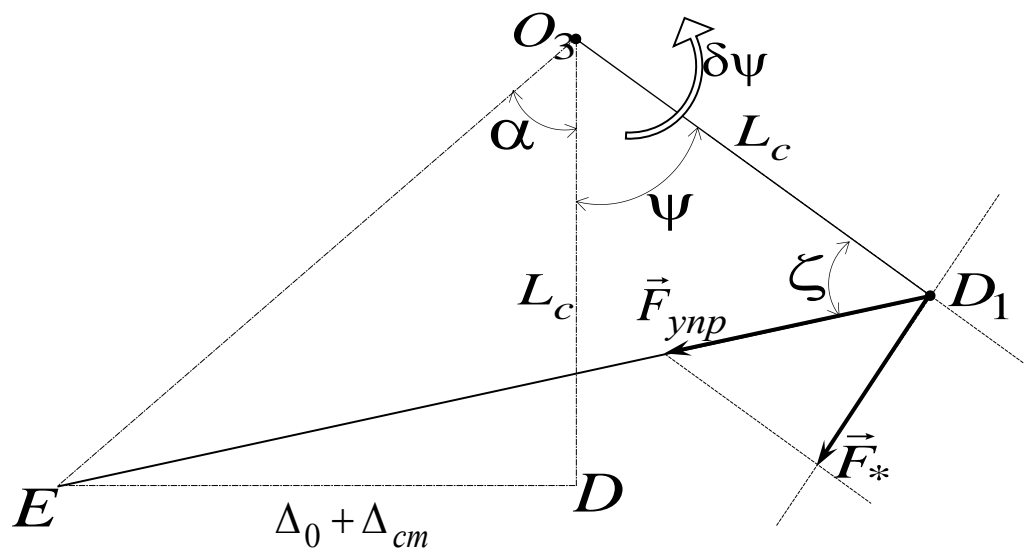


Рисунок 5. Робота пружної сили.

Очевидно, що $M_{\text{упр}} = F_* \cdot L_c = F_{\text{упр}} \cdot L_c \cdot \sin \zeta$.

Величину $\sin \zeta$ можна визначити з використанням теореми "синусів" для

трикутника EO_3D_1 :

$$\frac{ED_1}{\sin(\alpha + \psi)} = \frac{EO_3}{\sin \zeta},$$

звідки:
$$\sin \zeta = \frac{EO_3}{ED_1} \sin(\alpha + \psi) = \frac{EO_3}{ED_1} \cdot (\sin \alpha \cos \psi + \cos \alpha \sin \psi). \quad (24)$$

Недеформована пружина, коефіцієнт відновлення якої заданий і дорівнює c , має задану початкову довжину Δ_0 .

Статичне подовження Δ_{cm} цієї пружини, що забезпечує стан рівноваги тіла 3, зображений на Рисунку 2, можна обчислити з умови рівності нулю суми моментів сил, прикладених до цього тіла відносно центру O_3 :

$$c\Delta_{cm} \cdot L_0 = m_1 g \frac{r}{R} \cdot L, \quad \text{звідки} \quad \Delta_{cm} = \frac{m_1 g}{c} \cdot \frac{rL}{RL_c}, \quad (25)$$

де коефіцієнт $n_3 = L_c / L$ задається в межах: $0.6 \leq n_3 \leq 0.9$.

Таким чином, величина пружної сили, зображеної на Рисунку 5, дорівнюватиме $F_{упр} = c \cdot (ED_1 - \Delta_0)$.

Нам слід одержати вирази для довжин відрізків, що входять у наведені вище формули. Розглянемо трикутники EDO_3 и ED_1O_3 .

Маємо:

$$\begin{aligned} ED &= \Delta_0 + \Delta_{cm}, \quad O_3D = O_3D_1 = L_c, \quad (O_3E)^2 = (ED)^2 + (O_3D)^2, \\ \cos \alpha &= O_3D / O_3E = L_c / O_3E, \quad \sin \alpha = ED / O_3E = (\Delta_0 + \Delta_{cm}) / O_3E, \\ (ED_1)^2 &= (O_3E)^2 + (O_3D_1)^2 - 2 \cdot O_3E \cdot O_3D_1 \cdot \cos(\alpha + \psi) = \\ &= (O_3E)^2 + (O_3D_1)^2 - 2 \cdot O_3E \cdot O_3D_1 \cdot (\cos \alpha \cos \psi - \sin \alpha \sin \psi) = \\ &= (O_3E)^2 + (O_3D_1)^2 - 2 \cdot O_3D_1 \cdot (O_3D \cdot \cos \psi - ED \cdot \sin \psi) \end{aligned} \quad (26)$$

Виконаємо підстановки:

$$\begin{aligned} ED_1 &= \sqrt{(\Delta_0 + \Delta_{cm})^2 + 2L_c^2 - 2L_c(L_c \cos \psi - (\Delta_0 + \Delta_{cm}) \sin \psi)}, \quad (27) \\ \sin \zeta &= \frac{EO_3}{ED_1} \cdot (\sin \alpha \cos \psi + \cos \alpha \sin \psi) = \frac{(\Delta_0 + \Delta_{cm}) \cdot \cos \psi + L_c \cdot \sin \psi}{ED_1}. \end{aligned}$$

У підсумку момент пружної сили дорівнюватиме:

$$M_{\text{пруж}} = F_{\text{пруж}} \cdot L_c \cdot \sin \zeta = c \cdot \frac{(ED_1 - \Delta_0)}{ED_1} \cdot L_c \cdot [(\Delta_0 + \Delta_{cm}) \cdot \cos \psi + L_c \cdot \sin \psi].$$

Шукана узагальнена сила дорівнює $Q_{\text{пруж}} = -M_{\text{пруж}} \cdot \frac{1}{r} \frac{\mathbf{X}}{\mathbf{Z}}$, так як $\delta A = Q_{\text{пруж}} \cdot \delta \xi$.

Отже, остаточно маємо

$$Q_{\text{пруж}} = -M_{\text{пруж}} \cdot \frac{1}{r} \frac{\mathbf{X}}{\mathbf{Z}} = -c \cdot \frac{(ED_1 - \Delta_0)}{ED_1} \cdot \frac{L_c}{r} \cdot [(\Delta_0 + \Delta_{cm}) \cdot \cos \psi + L_c \cdot \sin \psi] \cdot \frac{\mathbf{X}}{\mathbf{Z}}.$$

З іншого боку, відомо, що пружна сила є потенційною, тобто

$$Q_{\text{пруж}} = -\frac{\partial \Pi_{\text{пруж}}}{\partial \xi},$$

де $\Pi_{\text{пруж}} = \frac{c}{2} [(ED_1 - \Delta_0)^2 - \Delta_{cm}^2]$ – потенційна енергія розглянутої пружини, що розглядається.

Очевидно, що

$$-\frac{\partial \Pi_{\text{пруж}}}{\partial \xi} = -c \cdot (ED_1 - \Delta_0) \frac{\partial ED_1}{\partial \xi}.$$

Скористаємося виразом (27):

$$\frac{\partial ED_1}{\partial \xi} = \frac{L_c (L_c \sin \psi + (\Delta_0 + \Delta_{cm}) \cos \psi)}{\sqrt{(\Delta_0 + \Delta_{cm})^2 + 2L_c^2 - 2L_c (L_c \cos \psi - (\Delta_0 + \Delta_{cm}) \sin \psi)}} \cdot \frac{\partial \psi}{\partial \xi},$$

і з урахуванням (16) отримаємо

$$\frac{\partial ED_1}{\partial \xi} = \frac{1}{ED_1} (L_c \sin \psi + (\Delta_0 + \Delta_{cm}) \cos \psi) \cdot \frac{L_c}{r} \frac{\mathbf{X}}{\mathbf{Z}}.$$

У підсумку ми отримали той самий результат для узагальненої сили $Q_{\text{пруж}}$, що свідчить на користь правильності виконаних математичних дій.

Тепер ми можемо повністю записати вираз для правої частини рівняння Лагранжа $-\frac{\partial(\Pi_1 + \Pi_3)}{\partial \xi} + Q_{yup}$, що містить потенційну енергію сил тяжіння, які

діють на механічну систему, та узагальнену силу Q_{yup} :

$$-\frac{\partial \Pi}{\partial \xi} = m_1 g - \frac{L}{2r} m_3 g \cdot \sin \psi \cdot \frac{\mathbf{X}}{\mathbf{Z}} - c \cdot \frac{ED_1 - \Delta_0}{ED_1} \cdot ((\Delta_0 + \Delta_{cm}) \cdot \cos \psi - L_c \cdot \sin \psi) \cdot \frac{L_c}{r} \frac{\mathbf{X}}{\mathbf{Z}}. \quad (30)$$

Для випадку **малих** коливань останній вираз істотно спрощується.

Горизонтальний зсув DD_1 точки D в положення D_1 можна вважати рівним $L_c \sin \psi$. Тоді потенційна енергія пружини буде

$$\Pi_{yup} = \frac{c}{2} [(DD_1 + \Delta_{cm})^2 - \Delta_{cm}^2], \text{ звідки з урахуванням співвідношення } \psi = \frac{1}{r} \frac{R}{L} \xi$$

отримаємо $DD_1 \approx L_c \psi = \frac{RL_c}{rL} \xi = \sigma \cdot \xi$, і тоді

$$\Pi_{yup} = \frac{c}{2} [(DD_1)^2 + 2DD_1 \cdot \Delta_{cm}] = \frac{c}{2} [(\sigma \xi)^2 + 2\sigma \xi \cdot \Delta_{cm}], \quad \frac{\partial \Pi_{yup}}{\partial \xi} = c [\sigma^2 \cdot \xi + \sigma \cdot \Delta_{cm}].$$

З урахуванням потенційних енергій сил тяжіння для тіл 1 і 3, а також скориставшись умовою рівноваги $m_1 g = c \cdot \sigma \Delta_{cm}$, що впливає зі співвідношення (25), отримаємо остаточно:

$$-\frac{\partial \Pi_*}{\partial \xi} = -\frac{\partial}{\partial \xi} [\Pi_1 + \Pi_3 + \Pi_{yup}] = m_1 g - \frac{m_3 g L}{2r} \xi - c \cdot \sigma^2 \xi - c \cdot \sigma \Delta_{cm} = -\left(c \sigma^2 + \frac{m_3 g L}{2r} \right) \xi. \quad (31)$$

У результаті рівняння лінійних коливань набуде вигляду:

$$\frac{1}{r^2} \left(J_1 + J_2 + J_3 \frac{R^2}{L^2} \right) \cdot \ddot{\xi} + \left(c \sigma^2 + \frac{m_3 g L}{2r} \right) \cdot \xi = 0,$$

$$\text{чи} \quad \ddot{\xi} + \omega_*^2 \cdot \xi = 0, \quad \omega_*^2 = \frac{\left(c \sigma^2 + \frac{m_3 g L}{2r} \right) r^2}{J_1 + J_2 + \frac{R^2}{L^2} J_3}, \quad (32)$$

де ω_* – кругова частота малих коливань розглянутої системи.

Закон малих коливань може бути записаний у формі:

$$\xi_*(t) = \xi_0 \cdot \cos(\omega_* t), \quad (33)$$

$$\text{коли початкові умови руху є } \xi(0) = \xi_0 \text{ и } \dot{\xi}(0) = 0. \quad (34)$$

Остаточний вигляд основного рівняння Лагранжа 2-го роду, що описує нелінійні коливання системи, і придатний для реалізації чисельного методу Рунге-Кутта, буде таким:

$$\begin{aligned} \frac{J_1 + J_2 + J_3 \cdot [\mathbf{X}/\mathbf{Z}]^2}{r^2} \cdot \ddot{\xi} = \\ = \frac{J_3}{r^3} \frac{\mathbf{X}}{\mathbf{Z}^3} \cdot \left[\mathbf{X} \cdot \left(\frac{\partial \mathbf{Z}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{Z}}{\partial \psi} \right) - \left(\frac{\partial \mathbf{X}}{\partial \varphi} + \frac{\mathbf{X}}{\mathbf{Z}} \cdot \frac{\partial \mathbf{X}}{\partial \psi} \right) \cdot \mathbf{Z} \right] \cdot \dot{\xi}^2 + \\ + m_1 g - \frac{m_3 g L}{2r} \cdot \sin \psi \frac{\mathbf{X}}{\mathbf{Z}} - c \cdot \frac{ED_1 - \Delta_0}{ED_1} \cdot ((\Delta_0 + \Delta_{cm}) \cdot \cos \psi + L_c \cdot \sin \psi) \cdot \frac{L_c}{r} \frac{\mathbf{X}}{\mathbf{Z}}. \end{aligned} \quad (35)$$

Особливість диференціального рівняння (35) полягає насамперед у тому, що воно містить вирази, в яких узагальнена координата ξ входить за допомогою неявних залежностей, що базуються на уявленнях виду (2), (6) і (17)-(19).

Така ситуація вимагає особливого підходу під час реалізації обчислень відповідно до класичної схеми чисельного методу Рунге-Кутта:

$$\begin{aligned} k_1 &= h \cdot f \left(t_j; \xi_j; \dot{\xi}_j \right), \\ k_2 &= h \cdot f \left(t_j + \frac{h}{2}; \xi_j + \frac{h}{2} \dot{\xi}_j + \frac{h}{8} k_1; \dot{\xi}_j + \frac{h}{2} k_1 \right), \\ k_3 &= h \cdot f \left(t_j + \frac{h}{2}; \xi_j + \frac{h}{2} \dot{\xi}_j + \frac{h}{8} k_1; \dot{\xi}_j + \frac{h}{2} k_2 \right), \\ k_4 &= h \cdot f \left(t_j + h; \xi_j + h \dot{\xi}_j + \frac{h}{2} k_3; \dot{\xi}_j + k_3 \right), \\ \dot{\xi}_{j+1} &= \dot{\xi}_j + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}, \quad \xi_{j+1} = \xi_j + h \cdot \left(\dot{\xi}_j + \frac{k_1 + k_2 + k_3}{6} \right), \end{aligned} \quad (36)$$

де $h = t_{j+1} - t_j$ – крок інтегрування за незалежною часовою змінною t .

Тут $f(t; \xi; \dot{\xi})$ – функція, за допомогою якої "обчислюється" значення правої частини інтегрованого диференціального рівняння (35), записаного у вигляді $\ddot{\xi} = f(t; \xi; \dot{\xi})$.

Як початкові умови для рівняння (35) беруться умови (34).

2.3 Аналіз механічної системи

Як уже зазначалося вище, розглянута механічна система перебуває під дією тільки консервативних сил і є положення стійкої рівноваги, що відповідає мінімальному значенню потенційної енергії.

Тому за початкових умов виду (34) слід було очікувати, що узагальнена координата ξ буде обмеженою функцією часу і $-\xi_0 \leq \xi(t) \leq \xi_0$.

Очевидно при цьому, що значення ξ_0 не може бути довільним, оскільки передбачуваний рух механічної системи має являти собою коливання біля її положення рівноваги.

Крім того, аналіз механічної системи з погляду її геометрії виявляє наявність двох "граничних" конфігурацій. Вибір такої конфігурації як початкових умов за нульових початкових швидкостей робить подальший рух тіл неоднозначним за напрямком і характером.

Розглянемо ці конфігурації.

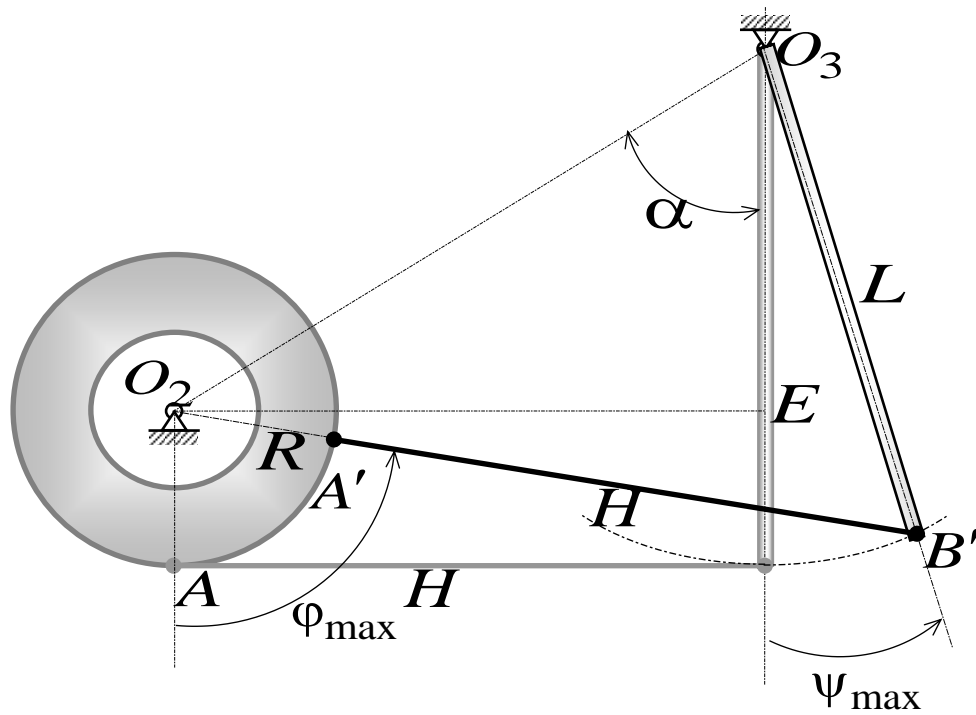


Рисунок 6. Перша гранична конфігурація системи.

Перша гранична конфігурація системи відповідає випадку, коли стрижень 3 відхиляється в позитивному напрямку відліку кута ψ на максимально можливий кут ψ_{\max} , а тіло 2 – повертається на кут φ_{\max} .

При цьому точка B стрижня AB досягає положення B' , яке відповідає найбільшому її віддаленню по горизонталі від положення рівноваги. Пряма $A'B'$ буде проходити через центр O_2 , тобто на цій прямій лежатиме радіус O_2A' .

Подальше збільшення кута φ призведе до зменшення кута ψ і зворотному руху точки B із положення B' по круговій траєкторії в напрямку положення рівноваги. Такий рух системи вже не буде коливанням, оскільки кут φ збільшуватиметься й надалі, а сама система не повертатиметься в положення рівноваги.

Граничне значення ψ_{\max} для кута ψ можна визначити з трикутника O_2O_3B' .

За теоремою косинусів: $(O_2B')^2 = (O_2O_3)^2 + (O_3B')^2 - 2 \cdot O_2O_3 \cdot O_3B' \cdot \cos(\alpha + \psi_{\max})$.

З прямокутного трикутника O_2O_3E знаходимо

$$O_2O_3 = \sqrt{(O_2E)^2 + (O_3E)^2} = \sqrt{H^2 + (L-R)^2}.$$

Оскільки $O_3B' = L$ и $O_2B' = H + R$, то

$$(H + R)^2 = H^2 + (L - R)^2 + L^2 - 2L \cdot \sqrt{H^2 + (L - R)^2} \cdot \cos(\alpha + \psi_{\max}),$$

звідки

$$\cos(\alpha + \psi_{\max}) = \frac{L^2 - R \cdot (L + H)}{L \cdot \sqrt{H^2 + (L - R)^2}}.$$

З прямокутного трикутника O_2O_3E можна визначити

$$\cos(\alpha) = \frac{L - R}{\sqrt{H^2 + (L - R)^2}}.$$

Остаточно отримуємо вираз для максимально допустимого початкового кута $\psi_0 < \psi_{\max}$:

$$\psi_0 < \psi_{\max} = \arccos \left[\frac{L^2 - R \cdot (L + H)}{L \cdot \sqrt{H^2 + (L - R)^2}} \right] - \arccos \left[\frac{L - R}{\sqrt{H^2 + (L - R)^2}} \right]. \quad (37)$$

Відповідно до цього, початковий кут повороту φ_0 тіла 2 і узагальнена початкова координата ξ_0 також матимуть відповідні максимально допустимі значення.

Другу граничну конфігурацію системи представлено на Рисунок 7.

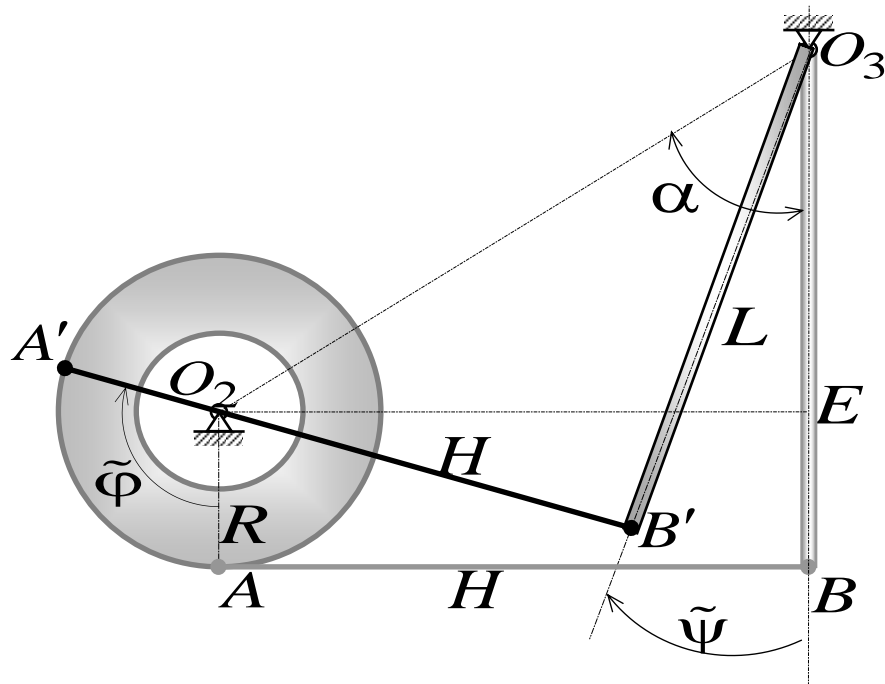


Рисунок 7. Друга гранична конфігурація системи.

За теоремою косинусів для трикутника O_2O_3B' маємо

$$(H - R)^2 = H^2 + (L - R)^2 + L^2 - 2L \cdot \sqrt{H^2 + (L - R)^2} \cdot \cos(\alpha + \tilde{\psi}),$$

звідки

$$\cos(\alpha + \tilde{\psi}) = \frac{L^2 - R \cdot (L - H)}{L \cdot \sqrt{H^2 + (L - R)^2}}.$$

Остаточно отримуємо вираз для мінімально можливого кута $\tilde{\psi} < 0$:

$$\tilde{\psi} = \arccos \left[\frac{L^2 - R \cdot (L - H)}{L \cdot \sqrt{H^2 + (L - R)^2}} \right] - \arccos \left[\frac{L - R}{\sqrt{H^2 + (L - R)^2}} \right] < 0.$$

Візуальний аналіз цієї конфігурації свідчить про те, що найменше допустиме значення кута $\tilde{\varphi} < 0$ за величиною виявляється більшим, ніж φ_{\max} з першої граничної конфігурації, тобто $|\tilde{\varphi}| > \varphi_{\max} > 0$.

Тому під час завдання конкретних початкових умов (34) для руху системи ми враховуватимемо й додаткову умову:

$$|\psi_0| < \psi_{\max}. \quad (38)$$

Аналіз діапазонів зміни кутів поворотів φ і ψ для тіл 2 і 3 можна виконати, розглянувши Рисунок 8.

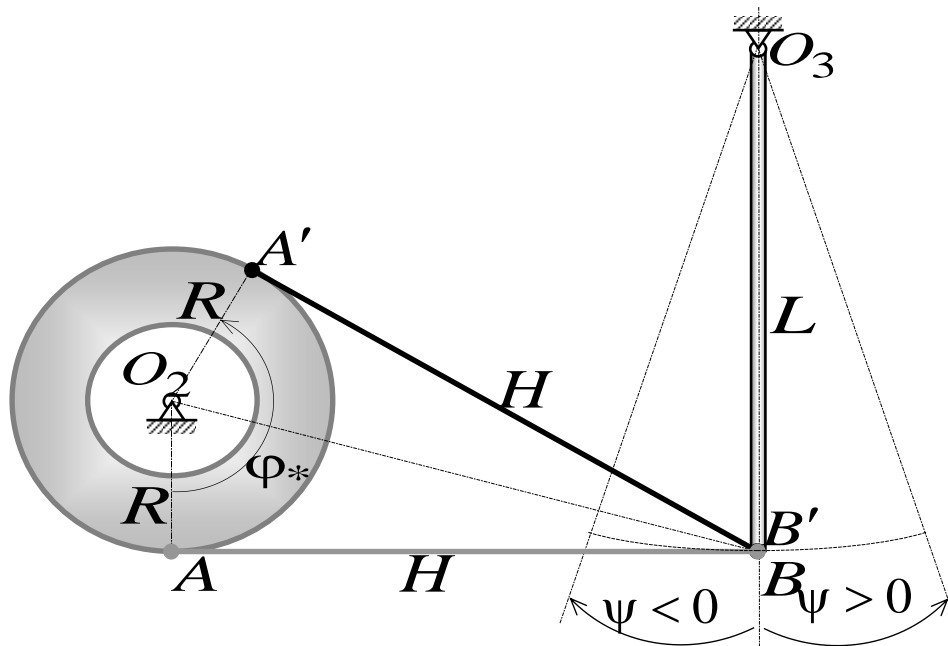


Рисунок 8. Проходження тіла 3 через вертикаль.

Тут зображено стрижень AB в положенні, при якому тіло 3 розташовується строго вертикально і $\psi = 0$.

Кут повороту тіла 2 при цьому дорівнюватиме $\varphi_* = 2 \cdot \arctg(H/R)$ (з прямокутного трикутника ABO_2).

Друге положення стрижня AB , при якому $\psi = 0$, відповідає положенню рівноваги, тобто при $\varphi = 0$ і при $\varphi = \varphi_*$ буде $\psi = 0$.

Таким чином, отримано відповідність між діапазонами кутів φ и ψ :

$$\text{при } 0 < \varphi < \varphi_* \text{ буде } 0 < \psi < \psi_{\max}; \quad \text{при } \varphi_* < \varphi < 2\pi \text{ буде } \tilde{\psi} < \psi < 0. \quad (39)$$

Треба зауважити, що для проведення чисельного експерименту слід установити діапазони допустимих значень для основних характеристик розглянутої механічної системи.

Пропонується така таблиця:

Таблиця 1

Маси тіл (кг)	Розміри тіл (м)	Параметри пружини
$2.00 \leq m_1 \leq 7.00$	$0.25 \leq r \leq 0.75$	$100.0 \leq c \leq 500.0$ (Н/м)
$1.00 \leq m_2 \leq 3.00$	$1.00 \leq R \leq 2.00$	$1.00 \leq \Delta_0 \leq 2.00$ (м)
$3.00 \leq m_3 \leq 9.00$	$2.00 \leq L \leq 5.00$	$0.60 \leq (L_0 / L) \leq 0.95$, $n_3 = L_0 / L$
	$3.00 \leq H \leq 9.00$	

Ще одне обмеження ми накладаємо на статичну деформацію Δ_{cm} пружини, що забезпечує положення спокою системи в заданій конфігурації (див. Рисунок 1).

Вимагаємо, щоб статична деформація не перевищувала 15% недеформованої довжини Δ_0 самої пружини.

Раніше ми отримали формулу для статичної деформації: $\Delta_{cm} = \frac{m_1 g}{c} \cdot \frac{r}{R} \cdot \frac{L_0}{L}$.

Отже, для задоволення умови

$$\Delta_{cm} \leq \frac{15}{100} \Delta_0, \quad (40)$$

ми можемо варіювати значення різних параметрів на свій розсуд.

Віддавати апріорі перевагу якомусь із них не видається обґрунтованим.

Сказане вище свідчить про те, що процес вибору початкових значень параметрів системи має супроводжуватися попередніми обчисленнями і наданням користувачеві інформації про виконання зазначених умов.

ВИБІР ПРОГРАМНОГО СЕРЕДОВИЩА ДЛЯ МОДЕЛЮВАННЯ

3.1 Критерії вибору оптимального програмного середовища

При ретельному відборі програмного середовища для розробки свого додатка, мої рішення були зумовлені кількома факторами:

1. Функціональний потенціал програмного середовища: важливо було, щоб воно надавало широкий набір інструментів, що спрощують процес розробки і забезпечують високу продуктивність.
2. Інтеграція з C#: можливість інтеграції обраного програмного середовища з мовою програмування C# та підтримка бібліотек і функцій, які планувалось використовувати.
3. Можливості моделювання: підтримка математичних операцій, розв'язання рівнянь руху та візуалізація результатів.
4. Легкість використання.

Враховуючи свій рівень знань та досвід з програмування та моделювання, для розробки застосунку було обране інтегроване середовище розробки Visual Studio та бібліотека компонентів графічного інтерфейсу користувача (GUI) C1.Win.C1Chart.4, розроблена компанією ComponentOne.

3.2 Опис обраного програмного середовища та його можливостей

Visual Studio є інтегроване середовище розробки (IDE) від компанії Microsoft, яке надає розширені засоби для розробки програмного забезпечення. Воно призначене для розробки різноманітних типів програм, включаючи десктопні програми, веб-додатки, мобільні додатки, хмарні рішення і багато іншого.

Має широкий спектр інструментів, які допомагають створювати, налагоджувати, тестувати і публікувати програми. Це включає інтегровані редактори коду для різних мов програмування, таких як C#, Visual Basic, C++, JavaScript, Python та інші.

Мій застосунок розроблявся з використанням потужної мови програмування C# та надійної бібліотеки C1.Win.C1Chart.4.

C1.Win.C1Chart.4 - це бібліотека компонентів графічного інтерфейсу користувача (GUI), розроблена компанією ComponentOne. Вона призначена для роботи з графіками та діаграмами в додатках, що розробляються мовою програмування C# з використанням платформи .NET.

Бібліотека C1.Win.C1Chart.4 надає розробникам можливість легко створювати та налаштовувати різні типи графіків і діаграм. Вона пропонує потужний набір функцій для візуалізації даних, масштабування, налаштування зовнішнього вигляду графіків та інших операцій, пов'язаних із графічним представленням інформації.

Саме ця бібліотека використовується в галузях, таких як наукові дослідження, фінанси, статистика, інженерія та інші, де важлива візуальна інтерпретація даних.

КОМП'ЮТЕРНА МОДЕЛЬ МЕХАНІЧНОЇ СИСТЕМИ

4.1 Огляд інтерфейсу та функціональності застосунку

Відповідно до поставленого завдання розроблено Windows-застосунок, що реалізує чисельний експеримент, під час якого визначаються кінематичні характеристики руху тіл заданої механічної системи.

Цей додаток містить низку сервісів, завданням яких є редагування основних фізичних параметрів механічної системи; обчислення на їхній основі параметрів малих коливань (періоду); аналіз кінематичної сумісності механічної системи.

Результати чисельного інтегрування рівняння Лагранжа 2-го роду, що описує динаміку заданої системи тіл, зберігаються у вигляді текстового файлу, структуру якого буде описано нижче, виводяться у вигляді таблиці на основну форму додатка і зображуються у вигляді графіків на відповідному елементі форми. Файл результатів зберігається безпосередньо в тій директорії, з якої здійснюється завантаження самого Windows-застосунку.

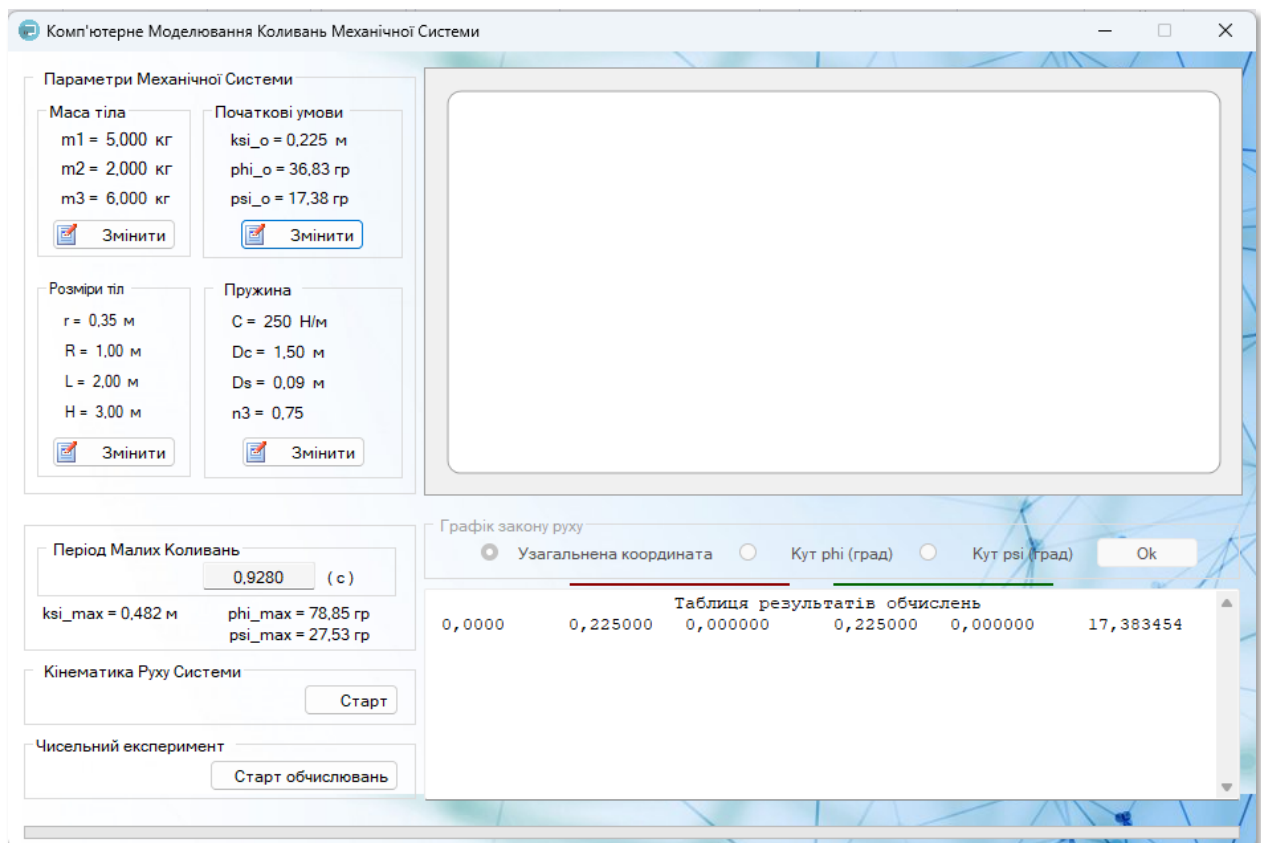


Рисунок 9. Основна форма застосунку

Параметри заданої механічної системи згруповано за фізичним змістом, що дало змогу реалізувати попередній аналіз даних, що вводяться.

Група "Маса тіла" призначена для перегляду значень і редагування мас усіх фізичних тіл, що входять до складу заданої механічної системи.

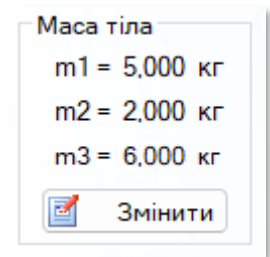


Рисунок 10. Маса тіл механічної системи

Під час натискання на кнопку "Змінити" відбувається ініціалізація допоміжної форми, призначеної для режиму редагування значень мас тіл системи.

Для виходу з режиму редагування зі збереженням нових значень для мас тіл системи слід натиснути на кнопку "Ок".

Якщо необхідно повернутися до "старих значень" редагованих параметрів - натиснути на кнопку "Відміна".

Для редагування значень кожної з мас тіл системи є окремий елемент керування. Крок зміни маси встановлено в 10 грамів.

Поточне значення маси тіла виводиться у верхній частині цього елемента.

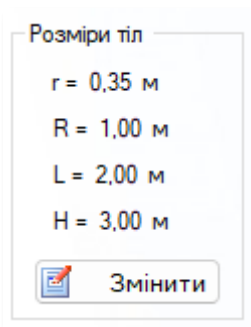
Діапазон зміни значень мас і одиниця виміру встановлені під час проектування цього додатка відповідно до *Таблиці 1*.

Зазначений підхід (редагування даних) реалізовано і для решти параметрів механічної системи і надалі на цьому ми не будемо зупинятися.

Інформація, розміщена внизу форми "Маса тіл механічної системи", призначена для контролю значення маси тіла 1, оскільки має місце обмеження (40) на величину заданого статичного подовження пружини.

Якщо користувач спробує перевищити граничне значення маси тіла 1, то буде видано відповідне попередження на цю форму.

У такий спосіб можна проводити чисельний експеримент для сукупностей "граничних" значень вхідних параметрів задачі та досліджувати відповідні рухи заданої механічної системи.



Група "Розміри тіл" призначена для перегляду значень і редагування розмірів (довжин) усіх фізичних тіл, що входять до складу заданої механічної системи.

Під час натискання на кнопку "Змінити" відбувається ініціалізація допоміжної форми, призначеної для режиму редагування розмірів тіл системи.

Діапазон зміни значень розмірів і одиниця їхнього виміру встановлені під час проектування цього додатка відповідно до *Таблиці 1*.

Крок зміни будь-якого розміру встановлено в 1 сантиметр.

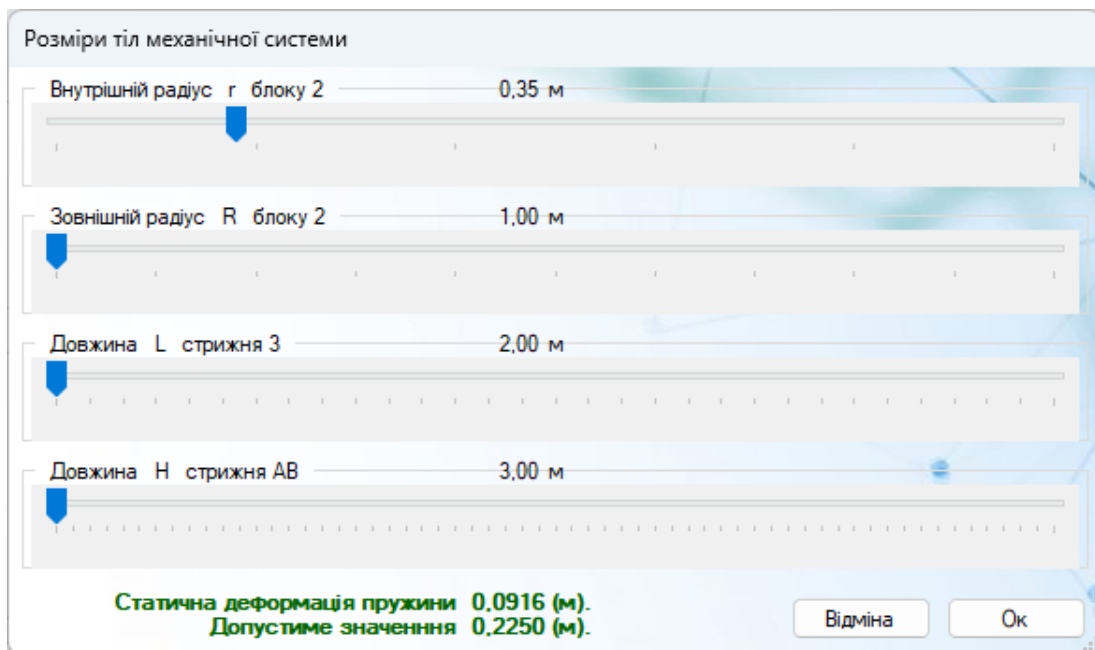


Рисунок 11. Розміри тіл механічної системи

Група "Пружина" призначена для перегляду значень і редагування параметрів пружини, а також положення точки її закріплення на стрижні 3.

Під час натискання на кнопку "Змінити" проводиться ініціалізація допоміжної форми, призначеної для режиму редагування параметрів пружини.

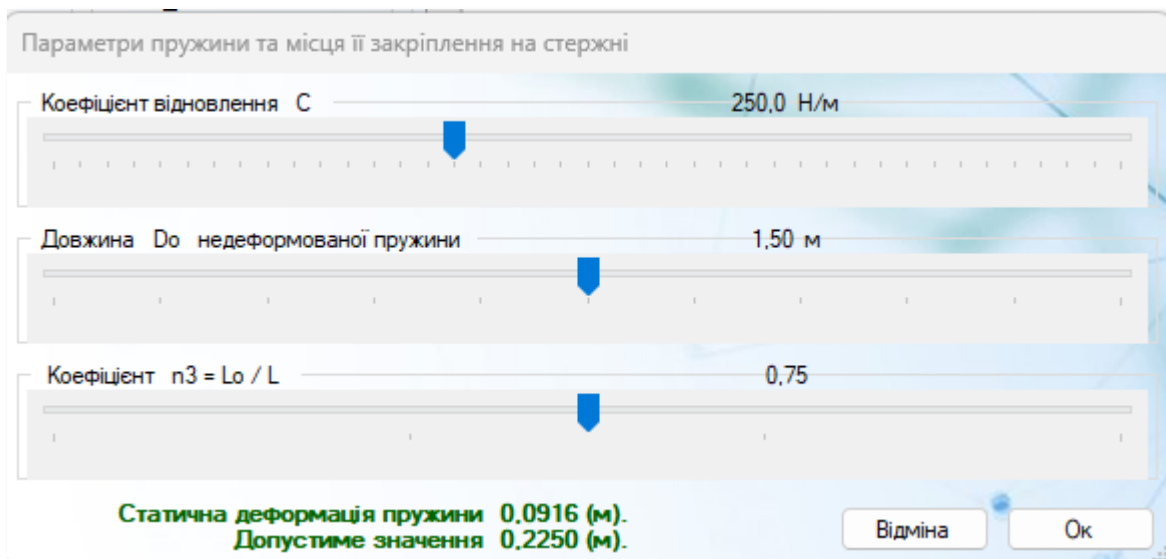
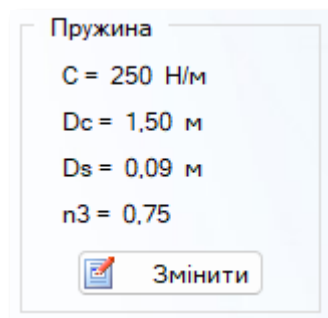
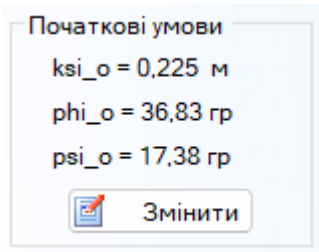


Рисунок 12. Параметри пружини та місця її закріплення на стрижні

Діапазон зміни значень параметрів та одиниці їх вимірювання встановлені під час проектування цього додатка відповідно до *Таблиці 1*.



Група "Початкова умова" призначена для перегляду значення і редагування початкового значення узагальненої координати.

Під час натискання на кнопку "Змінити" відбувається ініціалізація допоміжної форми, призначеної для задавання зазначеного значення. Початкова швидкість за умовою задачі дорівнює нулю (система починає рух зі стану спокою).

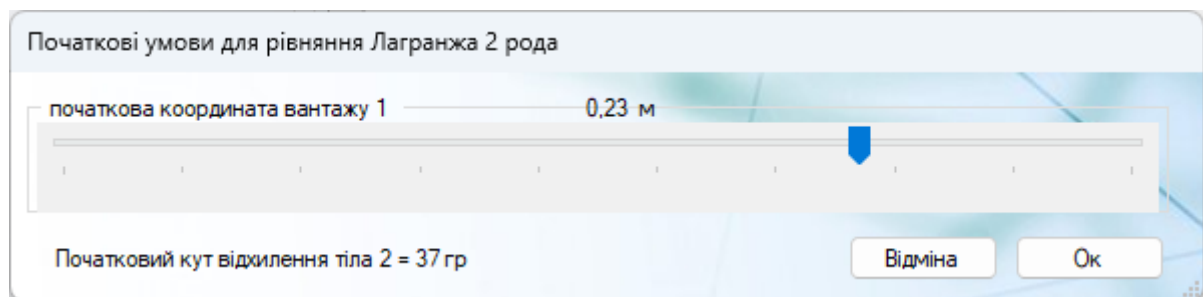
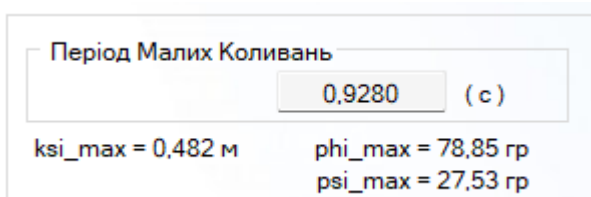


Рисунок 13. Початкові умови для рівняння Лагранжа 2 роду

Серед даних у групі "Початкова умова" знаходяться початкові значення кутів поворотів для тіл 2 і 3 відповідно. Це обчислювані, а не задані значення (оскільки механічна система має один ступінь свободи).



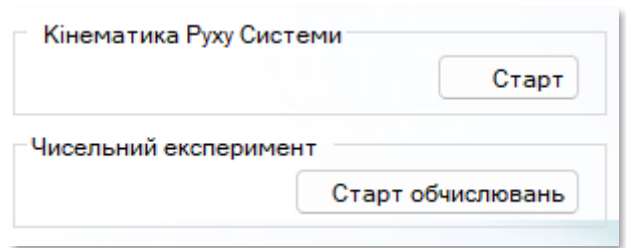
Наступна група на формі застосунку містить тільки довідкову інформацію.

По-перше, це - значення періоду для випадку "малих" коливань, тобто коли використовується (32) - лінеаризоване рівняння руху для заданої механічної системи.

По-друге, це гранично допустимі значення узагальненої координати і кутів поворотів тіл 2 і 3 механічної системи.

Знання цих величин необхідне для ухвалення рішення про початок обчислень за даним чисельним експериментом.

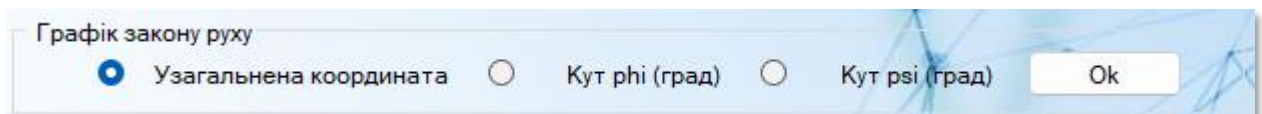
Група **"Кінематика руху системи"** призначена для ініціалізації допоміжної форми, на якій проводиться анімація кінематики механічної системи, що розглядається, без урахування діючих зовнішніх сил і моментів.



Група **"Чисельний експеримент"** призначена для ініціалізації розрахункової схеми Рунге-Кутта, кінцевою метою якої є розв'язання відповідної задачі Коші для ЧДВ 2-го порядку (рівняння Лагранжа 2-го роду)

Отримані результати зберігаються у вигляді текстового файлу і виводяться у вигляді таблиці та графіків на форму.

Перегляд графіка конкретної шуканої функції (після закінчення чисельного експерименту) здійснюється за допомогою радіо-кнопок, розташованих у групі **"Графік закону руху"**:



Результати чисельного моделювання руху заданої механічної системи являють собою таблицю з шести стовпчиків, див. Рисунок 14.

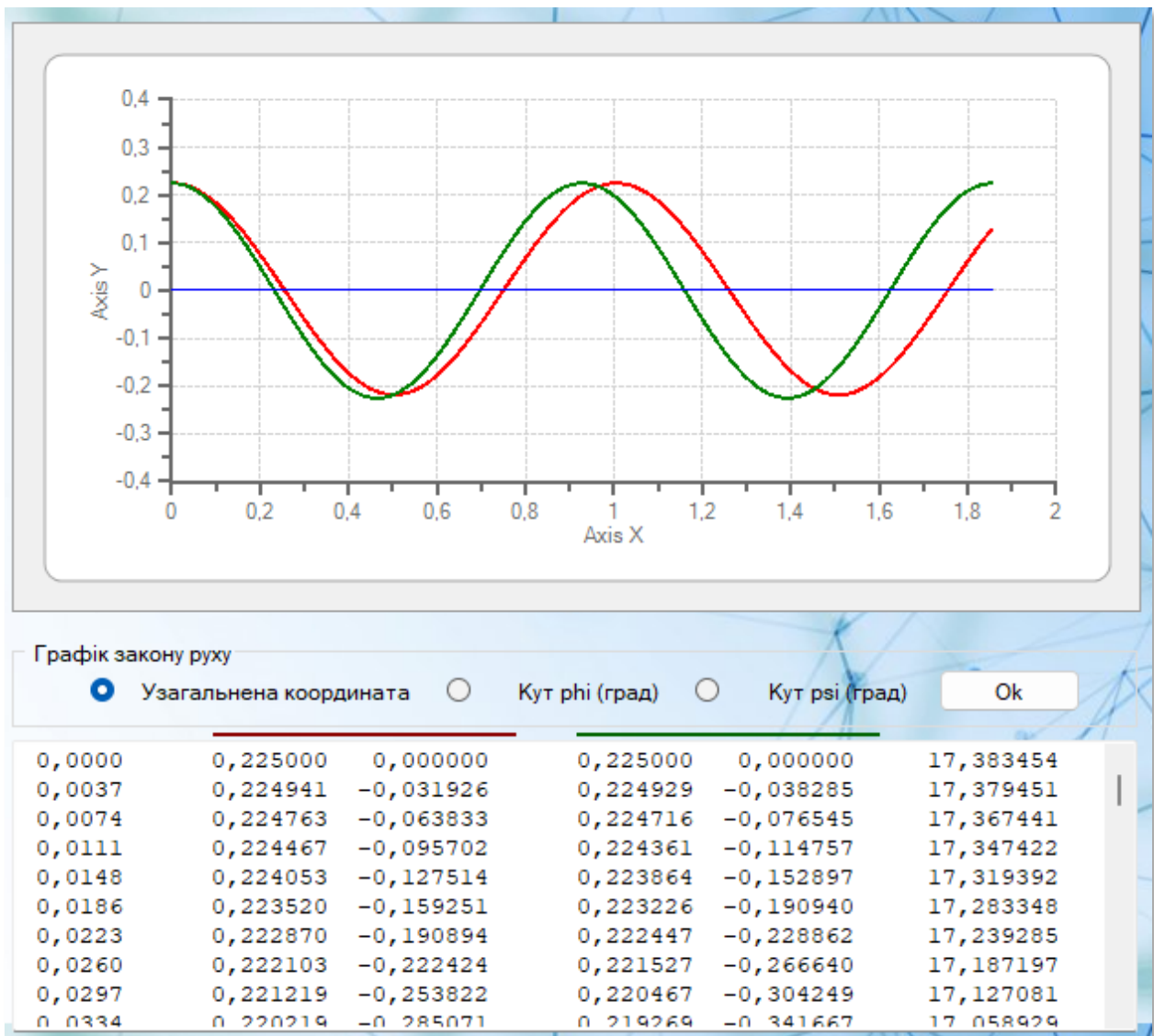


Рисунок 14. Результати чисельного експерименту

Перший стовпчик - послідовність моментів часу, для яких проводився розрахунок за методом Рунге-Кутта. Тому кожен рядок даних цієї таблиці відповідає тому моменту часу, що розташовується в першому стовпчику.

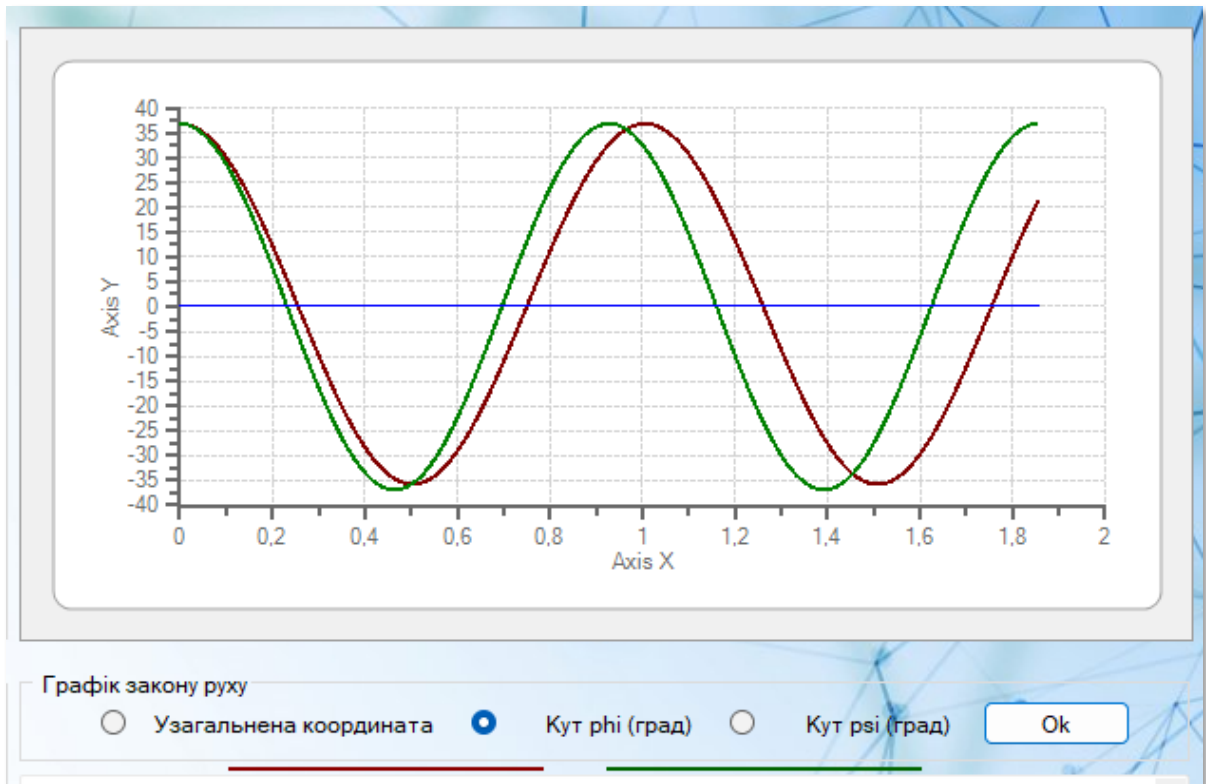
Другий і третій стовпці містять шукані величини узагальненої координати й узагальненої швидкості. Вони позначені вгорі червоною смужкою.

Четвертий і п'ятий стовпці - значення узагальненої координати і швидкості для випадку малих (лінійних) коливань. Вони позначені вгорі зеленою смужкою.

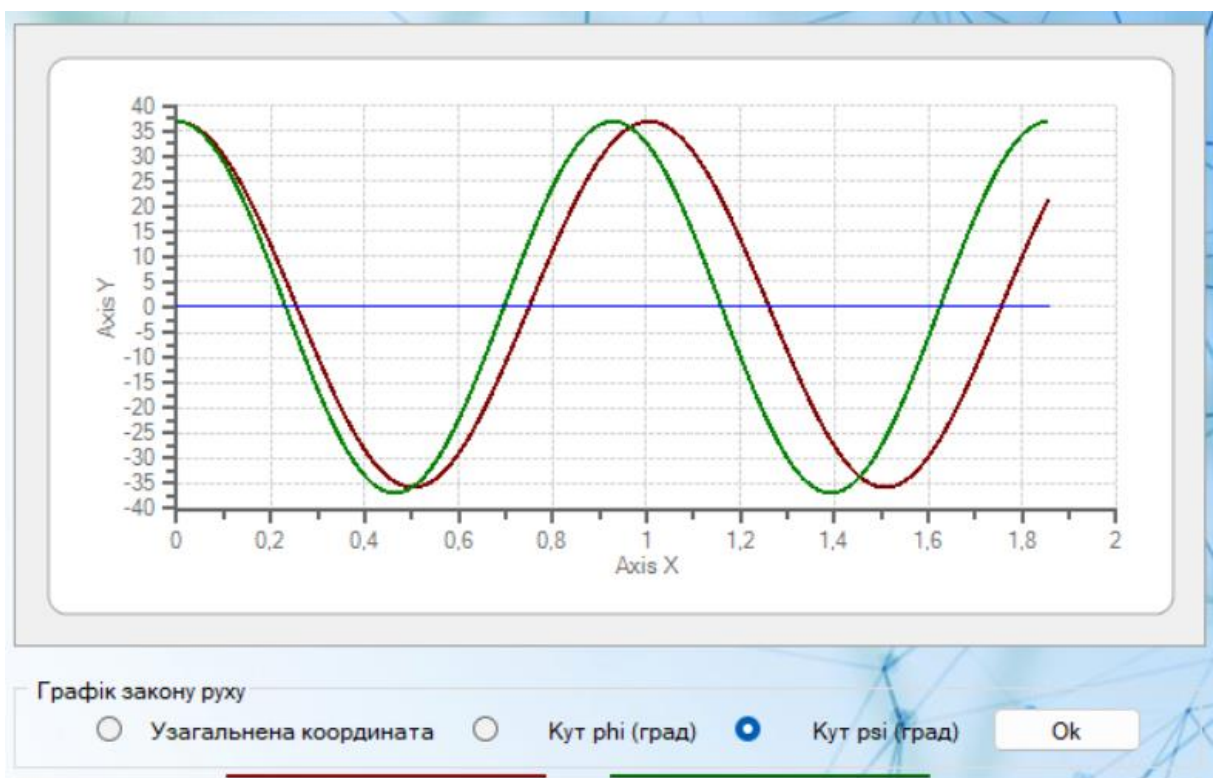
Останній стовпець - значення кута $\Psi = \Psi(\xi)$ для нелінійних коливань.

Цей самий принцип вибору кольорів збережено для графіків функцій: червоний (або темно червоний) - для нелінійних коливань, зелений - для функцій розв'язання лінеаризованої задачі:

для тіла 2 $\varphi = \varphi(t)$



для тіла 3 $\Psi = \Psi(t)$



Як уже зазначалося вище, натискання на кнопку "Старт" групи "Кінематика руху системи" ініціює допоміжну форму, на якій користувач може візуалізувати картину руху основних елементів механічної системи.

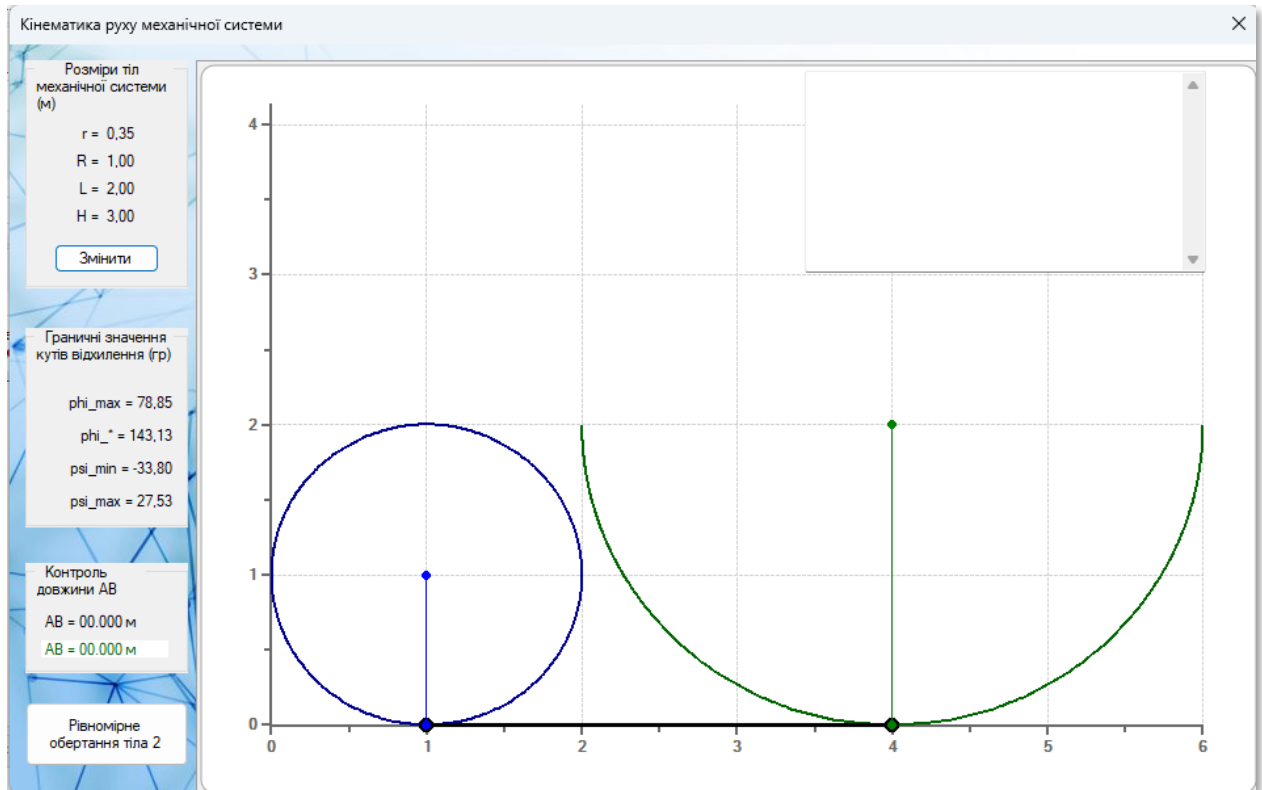


Рисунок 15. Кінематика руху механічної системи

Синім кольором зображено траєкторію точки, а зеленим кольором - точки.

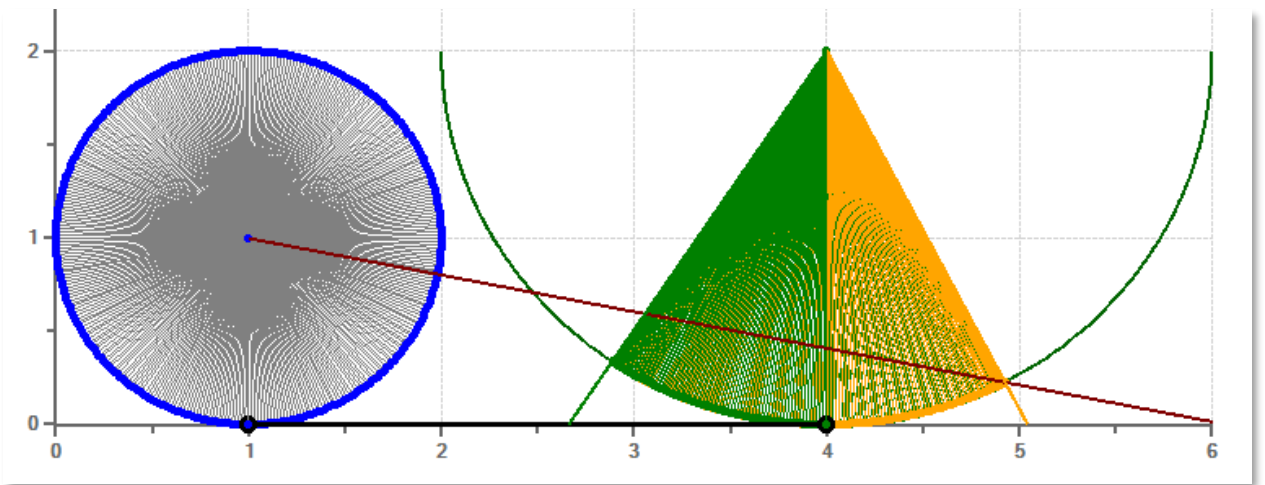
Відрізок чорного кольору, що з'єднує ці точки, зображує стрижень.

На цій формі є група "Розміри тіл механічної системи", яка призначена для редагування довжин. Однак маніпулювання розмірами в рамках цієї форми ніяк не позначається на даних чисельного експерименту і не переноситься на основну форму додатка.

Робота з цією формою вирішує завдання перевірки механічної системи на кінематичну реалізованість.

Користувач, змінюючи довжини елементів системи, може спостерігати їхню спільність у межах єдиного механічного механізму (конфігурацію).

Якщо конфігурація влаштовує користувача - він може побачити



картину руху механізму за умови рівномірного обертання тіла 2.

Рисунок 16. Кінематика руху механічної системи

На Рисунку 16 зображено фінальну картинку кінематики руху механічної системи. Це відповідає одному повному оберту тіла 2 навколо свого центру. Таким чином точка A опише повне коло (синього кольору).

При цьому точка B опише сектор своєї траєкторії, який складатиметься з двох фрагментів. Жовтий сектор відповідає позитивним значенням кута повороту $\psi = \psi(t)$, а зелений сектор - його від'ємним значенням $\psi = \psi(t)$.

Максимальне і мінімальне значення цього кута позначені відповідними променями, випущеними з центру траєкторії точки B .

Промінь червоного кольору визначає максимально можливу величину φ_{\max} для початкового значення кута повороту $\varphi = \varphi(t)$ тіла 2 (а з ним і граничного початкового значення ξ_0 узагальненої координати).

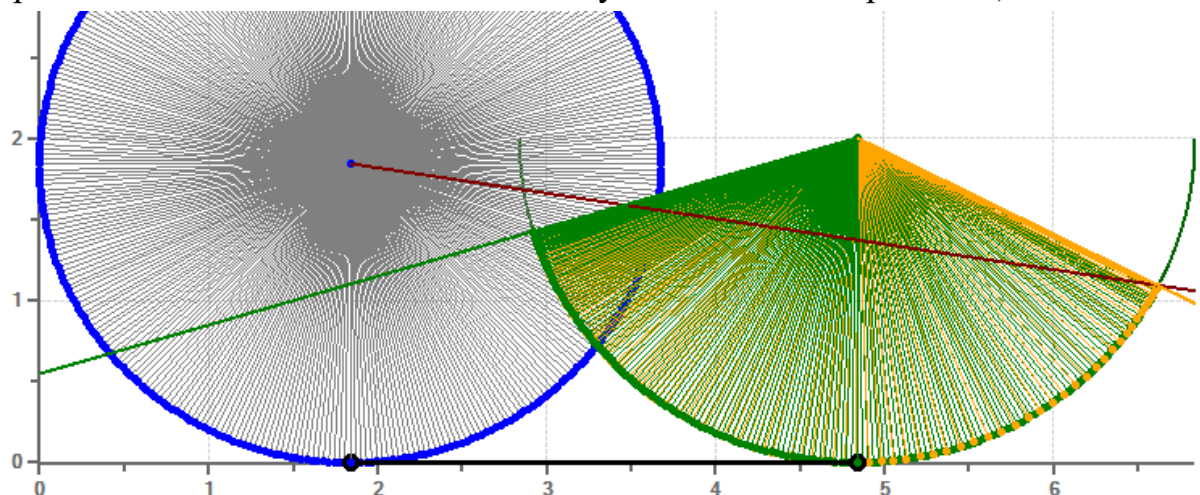


Рисунок 17. Кінематика руху механічної системи при $R \approx L$

4.2 Перевірка коректності моделі

Проведені експерименти показали, що найцікавішими з погляду кінематики руху є випадки, коли радіус тіла 2 майже дорівнює, але не перевершує розмір L стрижня 3, см. Рисунок 17.

Річ у тім, що за умови $R=L$ стрижень AB здійснюватиме суто поступальний рух, що є вже якісно іншим видом руху з точки зору кінематики. Тому коли $R \approx L$ спостерігається "нестійкість" зображення руху, пов'язана передусім із наявністю помилок округлення та впливом похибки методу дихотомії, що використовується для обчислень розв'язків рівняння (2).

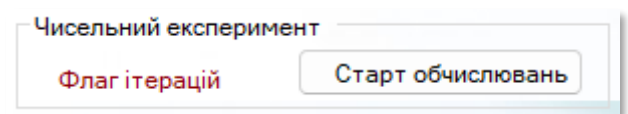
Ця ж "помилка" округлення в методі дихотомії проявляє себе на графіку функції $\psi = \psi(t)$ для випадку лінійних коливань, коли вхідні дані відповідають режиму, близькому до граничного для нелінійних коливань.

Дослідження окресленої проблеми виходить за межі цієї дипломної роботи, проте розроблена програма дасть змогу майбутнім користувачам її розв'язати.

Інша проблема чисельної нестійкості пов'язана з властивостями диференціального рівняння (35) і можливістю його однозначного інтегрування за певних початкових умов.

Виявлена нестійкість проявляє себе у вигляді "переходу" на інший режим руху системи, коли вона вже не здійснює коливання біля положення рівноваги, і при цьому обертання тіла 2 відбувається тільки в одному напрямку.

Такий режим обчислень супроводжується появою червоного рядка в групі "Чисельний експеримент".



Цей рядок невидимий у штатному режимі обчислень. У разі розвитку чисельної нестійкості цей рядок містить технічну інформацію про кількість ітерацій, витрачених на розрахунок однієї "точки" за методом Рунге-Кутта.

Застосунок, який було розроблено, можна вважати налагодженим і готовим для проведення досліджень. Проте, існує потенціал для його подальшого розширення.

АНАЛІЗ РЕЗУЛЬТАТІВ

Дослідження вільних коливань заданої механічної системи мало своєю основною метою виявити відмінність розв'язку лінеаризованої задачі від розв'язку, отриманого на основі нелінійного рівняння руху.

Очевидно, що за малих початкових відхилень системи від положення її рівноваги нелінійне рівняння Лагранжа має давати розв'язок (червоний колір графіка), близький до розв'язку лінеаризованого рівняння (зелений колір графіка).

Що і спостерігалось під час чисельних експериментів:

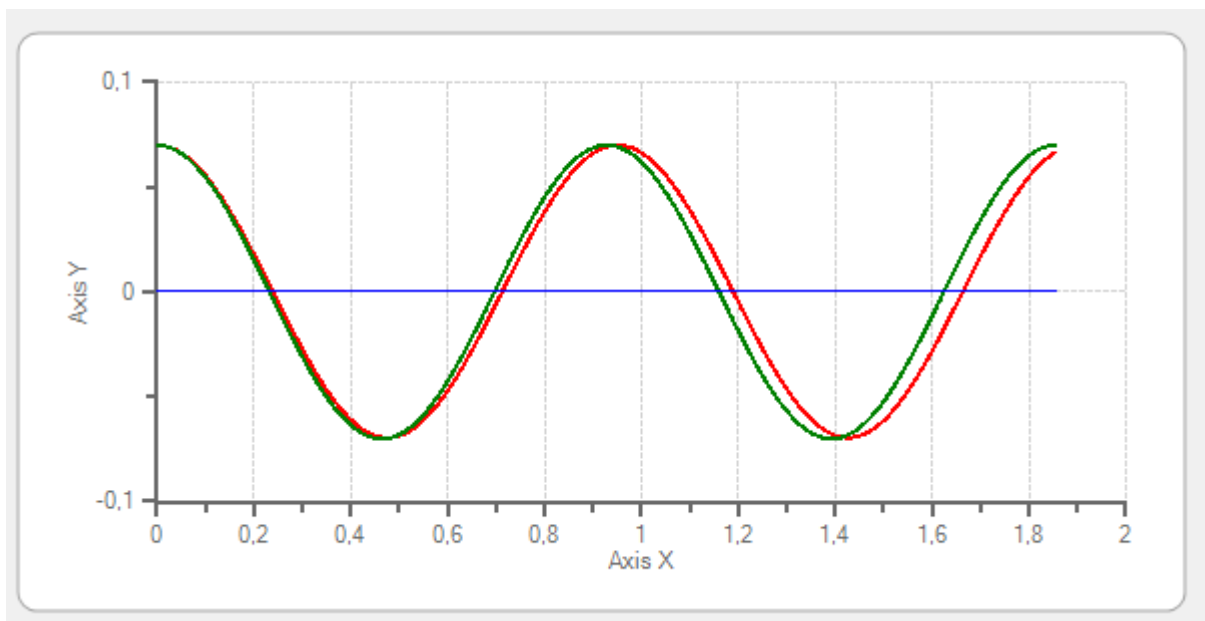


Рисунок 18. Залежність $\xi = \xi(t)$ при малому початковому значенні $\xi_0 = 0.07$ м.

На рисунку 18 можна спостерігати практичний збіг і амплітуд, і періодів розглянутих розв'язків, що свідчить про характер нелінійних коливань, близький до гармонійного.

Щойно початкове відхилення системи від положення рівноваги задається "скінченним", або іншими словами, його вже не можна вважати малим, розв'язки зазначених рівнянь істотно відрізняються.

Більше того, амплітуди нелінійного коливання для позитивної та негативної фази руху істотно відрізняються.

Відрізняються і часи тривалості цих фаз, див. Рисунки 19-20.

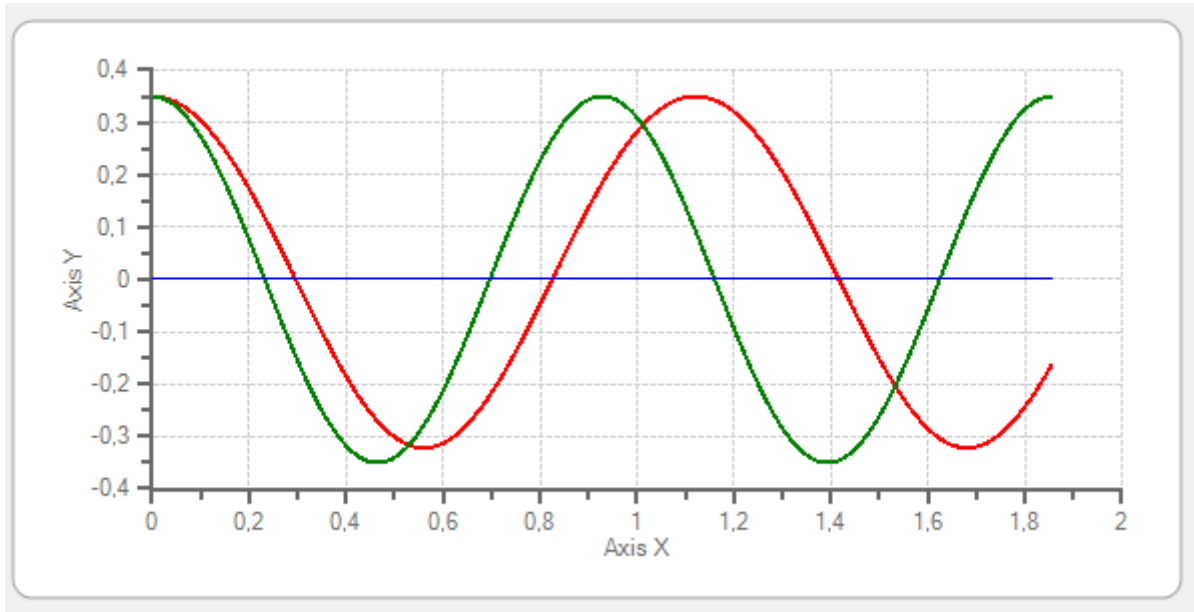


Рисунок 19. Залежність $\xi = \xi(t)$ при початковому значенні $\xi_0 = 0.35$ м.

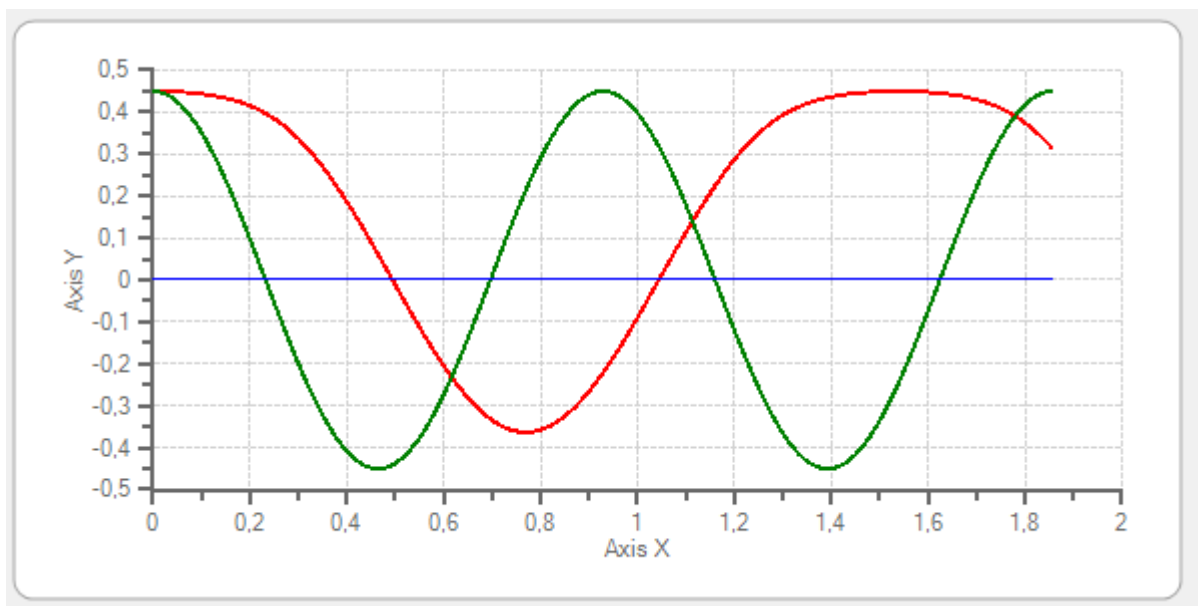


Рисунок 20. Залежність $\xi = \xi(t)$ при початковому значенні $\xi_0 = 0.45$ м.

На Рисунку 20 чітко видно, що нелінійні коливання мають період, майже на 50% більший, ніж період лінійних коливань.

Чисельні експерименти проводилися з різними комбінаціями значень вхідних параметрів і підтвердили зроблені вище висновки.

Таким чином, мету дипломної роботи можна вважати досягнутою, а завдання з чисельного моделювання - вирішеним у вигляді Windows-додатку.

ВИСНОВКИ

Отримано рівняння Лагранжа 2-го роду (35), що описує рух (нелінійні коливання) механічної системи з одним ступенем свободи.

При виведенні цього рівняння вираз для потенційної енергії діючої сили пружності виведено двома способами (з метою контролю правильності).

Знайдено розв'язок лінеаризованої задачі (32) про малі коливання системи біля положення її рівноваги.

Для розв'язання задачі Коші (рівняння Лагранжа 2-го роду) реалізовано чисельну схему Рунге-Кутта четвертого порядку.

Для розв'язання нелінійних алгебраїчних рівнянь виду (2) застосовували гібридні схеми на основі методу дихотомії та методу пошуку мінімуму для невід'ємної функції.

Досліджено кінематику можливих рухів заданої механічної системи.

Виявлено граничні значення для відхилень тіл системи, за яких можливі різні за характером подальші рухи цих тіл.

Створено Windows-застосунок, що реалізує чисельний експеримент за завданням дипломної роботи.

Цей додаток дає змогу користувачеві змінювати значення вхідних параметрів і проводити для новостворених наборів даних чисельний експеримент.

Результати експерименту зберігаються в текстовому файлі і можуть бути відтворені в будь-який час.

Є можливість візуального спостереження за кінематикою руху механічної системи за умови рівномірного обертання тіла 2.

Налагодження додатка виявило кілька причин, через які відбувається "чисельна" нестійкість.

Штатні обчислення показали суттєву відмінність розв'язку нелінійного рівняння механічних коливань для заданої системи тіл від розв'язку, отриманого для лінеаризованого рівняння.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кільчевський М.О. «Курс теоретичної механіки» Т. 1,2. – К: Київський університет, 2009. – 499с.
2. Мамаєв О.В., Нікулін О.В., Солод В. Ю. – Кам’янське: ДДТУ, 2018 – 247с
3. Булгаков В.М., Яременко В.В., Черниш О.М., Березовий М.Г. «Теоретична механіка»– К: Центр учбової літератури, 2021. – 640с.
4. Rob Miles «C# Programming Yellow Book» Edition 8.2 November 2016. – 216с.
5. Andrew Ttoelsen and Philip Japikse «Pro C# 7 With .NET and .Core» Eighth Edition by APress. Inc., Copyright © 2017, 1383с.
6. GrapeCity [Електронний ресурс]//Режим доступу:
<https://www.grapecity.com/componentone/winforms-ui-controls>
7. Microsoft.Learn [Електронний ресурс]//Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/csharp/>

ДОДАТКИ

ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ

MainForm

```

using System;
using System.Drawing;
using System.Threading;
using System.IO;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class MainForm : Form
    {
        private Form_Massa f_massa;
        private Form_Length f_length;
        private Form_Coil f_coil;
        private Form_Init f_init;
        private Form_Kinematics f_kinematics;

        public delegate double F_t_x_v(double t, double x, double v);

        public static double m1, m2, m3;
        public static double r, R, L, H, n3, Lc, Dc, C, Ds, DcDs;
        public static double omega, omega2, sigma, J1, J2, J3, r2, R2,
L2;
        public static double M1, M2, t_init, t_full, t_step, Psi_Ini;
        public static double Ksi_Max, Ksi_Ini, Phi_Max, Phi_Ini,
Psi_Max;
        public static double period_L, period_N;
        public static bool clean, start;

        public const double grad = 180.0 / Math.PI, g = 9.81, eps =
1.0E-8;
        public const float fgrad = (float)grad;
        public const double nC = 0.15;
        public const int Nt = 500, Mt = Nt + 1;

        public static float[] Ksi_Num, dKsi_Num, Ksi_An1, dKsi_An1,
Time;
        public static float[] Phi_Num, Phi_An1, Psi_An1, Psi_Num;

        public MainForm()
        {
            InitializeComponent(); clean = false; start = true;

            m1 = 5.0;
            m2 = 2.0;
            m3 = 6.0;

            r = 0.35;
            R = 1.00;
            L = 2.00;
            H = 3.00;

            C = 250.0;

```

```

Dc = 1.5;
n3 = 0.75;
c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();

Parameters(); start = false;
}

// Функция, вычисляющая вспомогательные параметры задачи

public void Parameters()
{
    double alfa, O2O3;

    label_m1.Text = "m1 = " + string.Format("{0:F3} кг",
m1);
    label_m2.Text = "m2 = " + string.Format("{0:F3} кг",
m2);
    label_m3.Text = "m3 = " + string.Format("{0:F3} кг",
m3);

    label_r1.Text = " r = " + string.Format("{0:F2} м", r);
    label_R2.Text = " R = " + string.Format("{0:F2} м", R);
    label_L.Text = " L = " + string.Format("{0:F2} м", L);
    label_H.Text = " H = " + string.Format("{0:F2} м", H);

    r2 = r * r; R2 = R * R; L2 = L * L; Lc = n3 * L;

    Ds = (m1 * g / C) * (r / R) * (L / Lc); // формула
(25)
    DcDs = Dc + Ds;

    label_C.Text = " C = " + string.Format("{0:F0} Н/м", C);
    label_Do.Text = " Dc = " + string.Format("{0:F2} м",
Dc);
    label_n3.Text = " n3 = " + string.Format("{0:F2} ",
n3);
    label_Ds.Text = " Ds = " + string.Format("{0:F2} м",
Ds);

    if ((nC * Dc) < Ds)
    {
        string err = " Статическая деформация пружины " +
string.Format("{0:F4} (м)\r\n", Ds) +
"превышает допустимое значение " +
string.Format("{0:F4} (м) !", Dc * nC);
        MessageBox.Show(err, " Ошибка исходных данных",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
        gBox_Run.Enabled = false;
    }
    else gBox_Run.Enabled = true;

    J1 = m1 * r2; // формулы (11)
    J2 = (m2 / 2.0) * (r2 * r2 + R2 * R2) / (r2 + R2);
    J3 = m3 * L2 / 3.0;

    M1 = J1 + J2 + J3 * R2 / L2; // формулы (32)

```

```

sigma = (R / r) * (Lc / L);
M2 = r2 * (C * sigma * sigma + 0.5 * m3 * g * L / r);
omega2 = M2 / M1; omega = Math.Sqrt(omega2);

period_L = 2.0 * Math.PI / omega;

tBx_T_0.Text = string.Format("{0:F4}", period_L);

t_full = 2.0 * period_L; t_step = t_full / Nt;

// Определяем предельно допустимые значения

O2O3 = Math.Sqrt(H * H + (L - R) * (L - R));
alfa = Math.Acos((L - R) / O2O3);
Psi_Max = Math.Acos((L * L - R * (L + H)) / L / O2O3) -
alfa;

    lbl_psi_max.Text = string.Format("psi_max = {0:F2} rp",
grad * Psi_Max);

    Phi_Max = Math.Acos((R * (L - H) - H * H - R * R) / (H +
R) / O2O3) - alfa;

    Ksi_Max = Phi_Max * r;

    lbl_phi_max.Text = string.Format("phi_max = {0:F2} rp",
grad * Phi_Max);
    lbl_ksi_max.Text = string.Format("ksi_max = {0:F3} м",
Ksi_Max);

// Понижаем предельное значение на 3 см
Ksi_Max = Math.Floor(Ksi_Max * 100 - 3) / 100.0;

if (start) Ksi_Ini = 0.5 * Ksi_Max;

if (Ksi_Ini > Ksi_Max) Ksi_Ini = Ksi_Max;

    lbl_ksi_o.Text = " ksi_o = " + string.Format("{0:F3} м",
Ksi_Ini);

    Phi_Ini = (Ksi_Ini / r) * grad;

    lbl_phi_o.Text = " phi_o = " + string.Format("{0:F2} rp",
Phi_Ini);

    Psi_Ini = PSI(Ksi_Ini) * grad;

    lbl_psi_o.Text = " psi_o = " + string.Format("{0:F2} rp",
Psi_Ini);
}

public static double PSI(double ksi)
{
    double a, b, c, fa, fc, fb; int k;

```

```

if (ksi == 0) return 0;
k = 0;
if (ksi > 0)
{
    a = 0; b = Psi_Max; fa = f(ksi, a);
    do
    {
        c = (a + b) * 0.5; fc = f(ksi, c); k++;
        if ((fa * fc) < 0) b = c; else a = c;
        if ((b - a) < eps) break;
    }
    while (Math.Abs(fc) > eps);
}
else
{
    a = -Psi_Max; b = 0; fb = f(ksi, b);
    do
    {
        c = (a + b) * 0.5; fc = f(ksi, c); k++;
        if ((fb * fc) < 0) a = c; else b = c;
        if ((b - a) < eps) break;
    }
    while (Math.Abs(fc) > eps);
}
return c;
}
// Функция, вычисляющая предельное значение Ksi (2)
public static double KSI(double psi)
{
    double d_ksi = 0.001, ksi = Ksi_Max, fa, fb = f(ksi, psi);
    do
    {
        d_ksi = -d_ksi / 10;
        do
        {
            fa = fb; ksi += d_ksi; fb = f(ksi, psi);
        }
        while (fa > fb);
    } while (fb > eps);
    return ksi;
}

// Функция, реализующая вычисления по формуле (1) или (2)
public static double f(double ksi, double psi)
{
    double phi = ksi / r;
    double x = R * (1.0 - Math.Cos(phi)) - L * (1.0 -
Math.Cos(psi));
    double y = H + L * Math.Sin(psi) - R * Math.Sin(phi);
    return x * x + y * y - H * H;
}

public static double fLinear(double t, double x, double v)
{
    return -omega2 * x;
}

```

```

}

public static double nonLinear(double t, double x, double v)
{
    double ksi, phi, psi, X_psi, X_phi, Z_psi, Z_phi, XZ, ED1;
    double X, Z, a, b, u1, u2, u3, u;

    ksi = x;

    // Определяем значения углов phi и psi

    phi = ksi / r;
    psi = PSI(ksi);

    double S_phi = Math.Sin(phi), S_psi = Math.Sin(psi),
           C_phi = Math.Cos(phi), C_psi = Math.Cos(psi),
           C_phs = Math.Cos(phi - psi);

    // формулы (6)

    a = R * (1.0 - C_phi) - L * (1.0 - C_psi);
    b = H + L * S_psi - R * S_phi;

    X = 2.0 * R * (a * S_phi - b * C_phi);
    Z = 2.0 * L * (a * S_psi - b * C_psi);

    XZ = X / Z;

    // формулы (19)

    X_phi = 2.0 * R * ((R - L) * C_phi + H * S_phi + L *
C_phs);
    X_psi = -2.0 * R * L * C_phs;
    Z_phi = 2.0 * R * L * C_phs;
    Z_psi = 2.0 * L * ((R - L) * C_psi + H * S_psi - R *
C_phs);

    u1 = J3 * XZ * (X * (Z_phi + XZ * Z_psi) - Z * (X_phi + XZ
* X_psi)) / Z / Z / r2 / r;

    u2 = m1 * g - 0.5 * m3 * g * S_psi * XZ * L / r;

    ED1 = Math.Sqrt(DcDs * DcDs + 2.0 * Lc * Lc - 2.0 * Lc *
(Lc * C_psi - DcDs * S_psi));
    u3 = C * (ED1 - Dc) / ED1 * (DcDs * C_psi + Lc * S_psi) *
Lc * XZ / r;

    u = (J1 + J2 + J3 * XZ * XZ) / r2;

    return (u1 * v * v + u2 - u3) / u;
}

private void btn_Calc_Click(object sender, EventArgs e)
{
    double x0, v0, x1, v1, xL, vL, t0, t1, xx; string txt =
"";
    btn_Calc.Enabled = false; btn_Kinematics.Enabled = false;

```

```

btn_coil.Enabled = false; btn_init.Enabled = false;
btn_length.Enabled = false; btn_massa.Enabled = false;
rBtn_ksi.Checked = true; gBx_Graph.Enabled = false;
this.Refresh(); Application.DoEvents();
Parameters();

#region <calc>

t1 = 0; x1 = Ksi_Ini; v1 = -0.000; xL = x1; vL = v1;
txt = string.Format("{0:F4}", t1).PadLeft(7) +
      string.Format("{0:F6}", x1).PadLeft(14) +
      string.Format("{0:F6}", v1).PadLeft(11) +
      string.Format("{0:F6}", xL).PadLeft(14) +
      string.Format("{0:F6}", vL).PadLeft(11) +
      string.Format("{0:F6}", Psi_Ini).PadLeft(14) +
"\r\n";

Ksi_An1 = new float[Mt]; dKsi_An1 = new float[Mt];
Ksi_Num = new float[Mt]; dKsi_Num = new float[Mt];
Time = new float[Mt];

Phi_An1 = new float[Mt]; Phi_Num = new float[Mt];
Psi_An1 = new float[Mt]; Psi_Num = new float[Mt];

Ksi_An1[0] = (float)Ksi_Ini; dKsi_An1[0] = 0;
Ksi_Num[0] = (float)Ksi_Ini; dKsi_Num[0] = 0; Time[0] = 0;
Phi_An1[0] = fgrad * Ksi_An1[0] / (float)r;
Phi_Num[0] = fgrad * Ksi_Num[0] / (float)r;
Psi_An1[0] = fgrad * (float)PSI(Ksi_Ini);
Psi_Num[0] = fgrad * (float)PSI(Ksi_Ini);

c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();

pBar.Maximum = Nt;

for (int i = 1; i <= Nt; i++)
{
    t0 = t1; x0 = x1; v0 = v1; t1 = t0 + t_step;

    //if (rBtn_1.Checked) RK2(t0, x0, v0, t1, ref x1, ref
v1, eps, fLinear);
    RK2(t0, x0, v0, t1, ref x1, ref v1, eps, nonLinear);

    xL = Ksi_Ini * Math.Cos(omega * t1);
    vL = -omega * Ksi_Ini * Math.Sin(omega * t1);

    xx = PSI(x1) * grad;

    txt += string.Format("{0:F4}", t1).PadLeft(7) +
           string.Format("{0:F6}", x1).PadLeft(14) +
           string.Format("{0:F6}", v1).PadLeft(11) +
           string.Format("{0:F6}", xL).PadLeft(14) +
           string.Format("{0:F6}", vL).PadLeft(11) +

```

```

                                string.Format("{0:F6}", xx).PadLeft(14) +
"\r\n";

                                Ksi_An1[i] = (float)xL; dKsi_An1[i] = (float)vL;
                                Ksi_Num[i] = (float)x1; dKsi_Num[i] = (float)v1;

                                Phi_An1[i] = fgrad * Ksi_An1[i] / (float)r; Phi_Num[i]
= fgrad * Ksi_Num[i] / (float)r;
                                Psi_An1[i] = fgrad * (float)PSI(xL); Psi_Num[i] =
fgrad * (float)PSI(x1);

                                Time[i] = (float)t1; pBar.Value = i;
Thread.Sleep(5); // pBar.Refresh();
                                }
                                tBx_table.Text = txt; pBar.Value = 0;

                                #endregion <calc>

                                #region <TheorMech.txt>

                                StreamWriter SW = new StreamWriter("TheorMech.txt");

                                txt = string.Format(" Масса тела 1  {0:F3} кг", m1);
SW.WriteLine(txt);
                                txt = string.Format(" Масса тела 2  {0:F3} кг", m2);
SW.WriteLine(txt);
                                txt = string.Format(" Масса тела 3  {0:F3} кг", m3);
SW.WriteLine(txt);
                                SW.WriteLine("");
                                txt = string.Format(" Радиус r тела 2  {0:F2} м", r);
SW.WriteLine(txt);
                                txt = string.Format(" Радиус R тела 2  {0:F2} м", R);
SW.WriteLine(txt);
                                txt = string.Format(" Длина L тела 3  {0:F2} м", L);
SW.WriteLine(txt);
                                txt = string.Format(" Длина AB          {0:F2} м", H);
SW.WriteLine(txt);
                                SW.WriteLine("");
                                txt = string.Format(" Жесткость пружины  {0:F0} ", C);
SW.WriteLine(txt);
                                txt = string.Format("          Длина пружины  {0:F2} ", Dc);
SW.WriteLine(txt);
                                txt = string.Format(" Деформация пружины  {0:F2} ", Ds);
SW.WriteLine(txt);
                                txt = string.Format("          Отношение Lo/L  {0:F2} ", n3);
SW.WriteLine(txt);
                                SW.WriteLine("");
                                txt = string.Format(" Начальное условие {0:F3} ",
Ksi_Ini); SW.WriteLine(txt);
                                SW.WriteLine("");
                                SW.WriteLine("      t          ksi          dksi          phi
psi");

                                for (int i = 0; i <= Nt; i++)
                                {
                                    txt = string.Format("{0:F4}", Time[i]).PadLeft(7) +
                                        string.Format("{0:F6}",
Ksi_Num[i]).PadLeft(14) +

```

```

        string.Format("{0:F6}",
dKsi_Num[i]).PadLeft(11) +
        string.Format("{0:F6}",
Phi_Num[i]).PadLeft(14) +
        string.Format("{0:F6}",
Psi_Num[i]).PadLeft(11);
        SW.WriteLine(txt);
    }
    SW.Close();

#endregion <TheorMech.txt>

#region <graph>

    C1.Win.C1Chart.ChartDataSeries sxN = new
C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxN);
    sxN.LineStyle.Color = Color.Red;
    sxN.LineStyle.Thickness = 2;
    sxN.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sxN.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
    sxN.Label = null;

    PointF[] dataN =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

    for (int i = 0; i <= Nt; i++)
    {
        dataN[i].X = Time[i];
        dataN[i].Y = Ksi_Num[i];
    }
    sxN.PointData.CopyDataIn(dataN);

    C1.Win.C1Chart.ChartDataSeries sxA = new
C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxA);
    sxA.LineStyle.Color = Color.Green;
    sxA.LineStyle.Thickness = 2;
    sxA.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sxA.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
    sxA.Label = null;

    PointF[] dataA =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

    for (int i = 0; i <= Nt; i++)
    {
        dataA[i].X = Time[i];
        dataA[i].Y = Ksi_An1[i];
    }
    sxA.PointData.CopyDataIn(dataA);

    C1.Win.C1Chart.ChartDataSeries s0 = new
C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s0);

```

```

        s0.LineStyle.Color = Color.Blue;
        s0.LineStyle.Thickness = 1;
        s0.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        s0.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
        s0.Label = null;

        PointF[] data0 =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

        data0[0].X = 0; data0[1].X = Time[Nt];
        data0[0].Y = 0; data0[1].Y = 0;
        s0.PointData.CopyDataIn(data0);

        #endregion <graph>

        Application.DoEvents();
        btn_Calc.Enabled = true; btn_Kinematics.Enabled = true;
        btn_coil.Enabled = true; btn_init.Enabled = true;
        btn_length.Enabled = true; gBx_Graph.Enabled = true;
        btn_massa.Enabled = true;
    }
private void
x1, RK2(double t0, double x0, double v0, double t1, ref double
ref double v1, double eps, F_t_x_v f)
{
    double k1, k2, k3, k4, ht, err, t, x, v, tj, xj, vj;
    int j, M, it = 0;
    // Определяем количество подинтервалов для нулевой
итерации
    M = 2 * (int)Math.Ceiling(t1 - t0);
    // Присваиваем "виртуальные" начальные значения
    x1 = 1.0E10; v1 = 1.0E10; x = 0.0; v = 0.0;
    do
    {
        ht = (t1 - t0) / M; it++;
        t = t0; x = x0; v = v0;
        for (j = 1; j <= M; j++)
        {
            tj = t;
            xj = x;
            vj = v;
            k1 = ht * f(t, x, v);

            t = tj + ht / 2;
            x = xj + ht * (vj + k1 / 4) / 2;
            v = vj + k1 / 2;
            k2 = ht * f(t, x, v);

            v = vj + k2 / 2;
            k3 = ht * f(t, x, v);

            t = tj + ht;
            x = xj + ht * (vj + k3 / 2);
            v = vj + k3;
            k4 = ht * f(t, x, v);

```

```

        x = xj + ht * (vj + (k1 + k2 + k3) / 6);
        v = vj + (k1 + 2 * (k2 + k3) + k4) / 6;
    }
    err = Math.Sqrt((x1 - x) * (x1 - x) + (v1 - v) * (v1 -
v));
    x1 = x; v1 = v; M = M * 2;
    if (it > 5)
    {
        this.lbl_calc.Text = string.Format("it = {0} M =
{1}", it, M);
        this.lbl_calc.Visible = true;
this.lbl_calc.Refresh();
    }
    }
    while ((err > eps) && (it < 14));
    return;
}
private void Cleaning()
{
    if (clean)
    {
        c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();
        tBx_table.Text = ""; lbl_calc.Visible = false; clean =
false;
    }
}
private void btn_massa_Click(object sender, EventArgs e)
{
    f_massa = new Form_Massa(); f_massa.Activate();
f_massa.ShowDialog();
    Parameters(); Cleaning();
}
private void btn_length_Click(object sender, EventArgs e)
{
    f_length = new Form_Length(); f_length.Activate();
f_length.ShowDialog();
    Parameters(); Cleaning();
}
private void btn_coil_Click(object sender, EventArgs e)
{
    f_coil = new Form_Coil(); f_coil.Activate();
f_coil.ShowDialog();
    Parameters(); Cleaning();
}
private void btn_init_Click(object sender, EventArgs e)
{
    f_init = new Form_Init(); f_init.Activate();
f_init.ShowDialog();
    Parameters(); Cleaning();
}
private void btn_Graf_Click(object sender, EventArgs e)
{
    c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();

    if (rBtn_ksi.Checked)
    {

```

```

        C1.Win.C1Chart.ChartDataSeries sxN = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxN);
    sxN.LineStyle.Color = Color.Red;
    sxN.LineStyle.Thickness = 2;
    sxN.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sxN.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
    sxN.Label = null;

    PointF[] dataN =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

    for (int i = 0; i <= Nt; i++)
    {
        dataN[i].X = Time[i];
        dataN[i].Y = Ksi_Num[i];
    }
    sxN.PointData.CopyDataIn(dataN);

        C1.Win.C1Chart.ChartDataSeries sxA = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxA);
    sxA.LineStyle.Color = Color.Green;
    sxA.LineStyle.Thickness = 2;
    sxA.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sxA.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
    sxA.Label = null;

    PointF[] dataA =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

    for (int i = 0; i <= Nt; i++)
    {
        dataA[i].X = Time[i];
        dataA[i].Y = Ksi_An1[i];
    }
    sxA.PointData.CopyDataIn(dataA);

        C1.Win.C1Chart.ChartDataSeries s0 = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s0);
    s0.LineStyle.Color = Color.Blue;
    s0.LineStyle.Thickness = 1;
    s0.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    s0.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
    s0.Label = null;

    PointF[] data0 =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

    data0[0].X = 0; data0[1].X = Time[Nt];

```

```

        data0[0].Y = 0; data0[1].Y = 0;
        s0.PointData.CopyDataIn(data0);
    }

    if (rBtn_phi.Checked)
    {
        C1.Win.C1Chart.ChartDataSeries sxN = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxN);
        sxN.LineStyle.Color = Color.Maroon;
        sxN.LineStyle.Thickness = 2;
        sxN.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        sxN.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
        sxN.Label = null;

        PointF[] dataN =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

        for (int i = 0; i <= Nt; i++)
        {
            dataN[i].X = Time[i];
            dataN[i].Y = Phi_Num[i];
        }
        sxN.PointData.CopyDataIn(dataN);

        C1.Win.C1Chart.ChartDataSeries sxA = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxA);
        sxA.LineStyle.Color = Color.Green;
        sxA.LineStyle.Thickness = 2;
        sxA.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        sxA.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
        sxA.Label = null;

        PointF[] dataA =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

        for (int i = 0; i <= Nt; i++)
        {
            dataA[i].X = Time[i];
            dataA[i].Y = Phi_An1[i];
        }
        sxA.PointData.CopyDataIn(dataA);

        C1.Win.C1Chart.ChartDataSeries s0 = new
C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s0);
        s0.LineStyle.Color = Color.Blue;
        s0.LineStyle.Thickness = 1;
        s0.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        s0.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
    }

```

```

        s0.Label = null;

        PointF[] data0 =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

        data0[0].X = 0; data0[1].X = Time[Nt];
        data0[0].Y = 0; data0[1].Y = 0;
        s0.PointData.CopyDataIn(data0);
    }

    if (rBtn_psi.Checked)
    {

        C1.Win.C1Chart.ChartDataSeries sxA = new
C1.Win.C1Chart.ChartDataSeries();

        c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxA);
        sxA.LineStyle.Color = Color.Green;
        sxA.LineStyle.Thickness = 2;
        sxA.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        sxA.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
        sxA.Label = null;

        PointF[] dataA =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

        for (int i = 0; i <= Nt; i++)
        {
            dataA[i].X = Time[i];
            dataA[i].Y = -Psi_An1[i];
        }
        sxA.PointData.CopyDataIn(dataA);

        C1.Win.C1Chart.ChartDataSeries sxB = new
C1.Win.C1Chart.ChartDataSeries();

        c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sxB);
        sxB.LineStyle.Color = Color.Brown;
        sxB.LineStyle.Thickness = 2;
        sxB.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        sxB.FitType = C1.Win.C1Chart.FitTypeEnum.Spline;
        sxB.Label = null;

        PointF[] dataN =
(PointF[])Array.CreateInstance(typeof(PointF), Mt);

        for (int i = 0; i <= Nt; i++)
        {
            dataN[i].X = Time[i];
            dataN[i].Y = Psi_Num[i];
        }
        sxB.PointData.CopyDataIn(dataN);

        C1.Win.C1Chart.ChartDataSeries s0 = new
C1.Win.C1Chart.ChartDataSeries();
    }
}

```

```

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s0);
    s0.LineStyle.Color = Color.Blue;
    s0.LineStyle.Thickness = 1;
    s0.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    s0.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
    s0.Label = null;

    PointF[] data0 =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

    data0[0].X = 0; data0[1].X = Time[Nt];
    data0[0].Y = 0; data0[1].Y = 0;
    s0.PointData.CopyDataIn(data0);
    }
    }
private void rBtn_ksi_CheckedChanged(object sender, EventArgs
e)
    {
        c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();
    }
private void rBtn_phi_CheckedChanged(object sender, EventArgs
e)
    {
        c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();
    }
private void rBtn_psi_CheckedChanged(object sender, EventArgs
e)
    {
        c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();
    }
private void btn_Kinematics_Click(object sender, EventArgs e)
    {
        f_kinematics = new Form_Kinematics();
f_kinematics.Activate();
        f_kinematics.ShowDialog();
    }
    }
}

```

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ Form_Massa

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Massa : Form
    {
        private static double m1_min, m1_max, m1;
        private static double m2_min, m2_max, m2;
        private static double m3_min, m3_max, m3;
        private static double d_stat, d, d_max;

        public Form_Massa()
        {
            InitializeComponent();

            m1_min = 2.0; m1_max = 7.0;
            m1 = MainForm.m1;
            Bar_m1.Maximum = (int)(100 * m1_max - 100 * m1_min);
            Bar_m1.Value = (int)(Bar_m1.Maximum * (m1 - m1_min) /
(m1_max - m1_min));
            lbl_m1.Text = string.Format("{0:F3} кг", m1);

            m2_min = 1.0; m2_max = 3.0;
            m2 = MainForm.m2;
            Bar_m2.Maximum = (int)(100 * m2_max - 100 * m2_min);
            Bar_m2.Value = (int)(Bar_m2.Maximum * (m2 - m2_min) /
(m2_max - m2_min));
            lbl_m2.Text = string.Format("{0:F3} кг", m2);

            m3_min = 3.0; m3_max = 9.0;
            m3 = MainForm.m3;
            Bar_m3.Maximum = (int)(100 * m3_max - 100 * m3_min);
            Bar_m3.Value = (int)(Bar_m3.Maximum * (m3 - m3_min) /
(m3_max - m3_min));
            lbl_m3.Text = string.Format("{0:F3} кг", m3);

            d = MainForm.g * MainForm.r * MainForm.L / (MainForm.C *
MainForm.R * MainForm.Lc);
            d_stat = MainForm.m1 * d;
            d_max = 0.15 * MainForm.Dc;

            if (d_stat > d_max)
            {
                label_alarm.Text = "Статична деформація пружини " +
string.Format("{0:F4} (м)\r\n",
d_stat) +
                " перевищує допустиме значення " +
                string.Format("{0:F4} (м) !",
d_max);
                label_alarm.ForeColor = Color.DarkRed;
            }
            else

```

```

        {
            label_alarm.Text = "Статична деформація пружини " +
                string.Format("{0:F4} (м).\r\n",
d_stat) +
                " Допустиме значення " +
                string.Format("{0:F4} (м).",
d_max);
            label_alarm.ForeColor = Color.DarkGreen;
        }
    }

private void Bar_m1_Scroll(object sender, EventArgs e)
{
    int m = Bar_m1.Value;
    m1 = m1_min + m * (m1_max - m1_min) / Bar_m1.Maximum;
    lbl_m1.Text = string.Format("{0:F3} кг", m1);

    d_stat = m1 * d;

    if (d_stat > d_max)
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м)\r\n",
d_stat) +
            " перевищує допустиме значення "
+
            string.Format("{0:F4} (м) !",
d_max);
        label_alarm.ForeColor = Color.DarkRed;
    }
    else
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м).\r\n",
d_stat) +
            " Допустиме значення " +
            string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void Bar_m2_Scroll(object sender, EventArgs e)
{
    int m = Bar_m2.Value;
    m2 = m2_min + m * (m2_max - m2_min) / Bar_m2.Maximum;
    lbl_m2.Text = string.Format("{0:F3} кг", m2);
}

private void Bar_m3_Scroll(object sender, EventArgs e)
{
    int m = Bar_m3.Value;
    m3 = m3_min + m * (m3_max - m3_min) / Bar_m3.Maximum;
    lbl_m3.Text = string.Format("{0:F3} кг", m3);
}

private void button1_Click(object sender, EventArgs e)
{

```

```
        MainForm.clean = false; this.Close();
    }

    private void btn_Ok_Click(object sender, EventArgs e)
    {
        MainForm.m1 = m1; MainForm.m2 = m2; MainForm.m3 = m3;
        MainForm.clean = true; this.Close();
    }
}
}
```

ДОДАТОК В. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ Form_Length_K

```

using System;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Length_K : Form
    {
        private static double r_min, r_max, r;
        private static double R_min, R_max, R;
        private static double L_min, L_max, L;
        private static double H_min, H_max, H;
        public Form_Length_K()
        {
            InitializeComponent();

            r_min = 0.25; r_max = 0.75;
            r = Form_Kinematics.r;
            Bar_r1.Maximum = (int)(100 * r_max - 100 * r_min);
            Bar_r1.Value = (int)(Bar_r1.Maximum * (r - r_min) / (r_max
- r_min));
            lbl_r1.Text = string.Format("{0:F2} м", r);

            R_min = 1.00; R_max = 3.00;
            R = Form_Kinematics.R;
            Bar_r2.Maximum = (int)(100 * R_max - 100 * R_min);
            Bar_r2.Value = (int)(Bar_r2.Maximum * (R - R_min) / (R_max
- R_min));
            lbl_r2.Text = string.Format("{0:F2} м", R);

            L_min = 2.00; L_max = 7.00;
            L = Form_Kinematics.L;
            Bar_L.Maximum = (int)(100 * L_max - 100 * L_min);
            Bar_L.Value = (int)(Bar_L.Maximum * (L - L_min) / (L_max -
L_min));
            lbl_L.Text = string.Format("{0:F2} м", L);

            H_min = 2.00; H_max = 9.00;
            H = Form_Kinematics.H;
            Bar_H.Maximum = (int)(100 * H_max - 100 * H_min);
            Bar_H.Value = (int)(Bar_H.Maximum * (H - H_min) / (H_max -
H_min));
            lbl_H.Text = string.Format("{0:F2} м", H);
        }
        private void Bar_r1_Scroll(object sender, EventArgs e)
        {
            int m = Bar_r1.Value;
            r = r_min + m * (r_max - r_min) / Bar_r1.Maximum;
            lbl_r1.Text = string.Format("{0:F2} м", r);
        }
        private void Bar_r2_Scroll(object sender, EventArgs e)
        {
            int m = Bar_r2.Value;
            R = R_min + m * (R_max - R_min) / Bar_r2.Maximum;
            lbl_r2.Text = string.Format("{0:F2} м", R);
        }
    }
}

```

```
private void Bar_L_Scroll(object sender, EventArgs e)
{
    int m = Bar_L.Value;
    L = L_min + m * (L_max - L_min) / Bar_L.Maximum;
    lbl_L.Text = string.Format("{0:F2}  m", L);
}
private void Bar_H_Scroll(object sender, EventArgs e)
{
    int m = Bar_H.Value;
    H = H_min + m * (H_max - H_min) / Bar_H.Maximum;
    lbl_H.Text = string.Format("{0:F2}  m", H);
}
private void btn_Exit_Click(object sender, EventArgs e)
{
    this.Close();
}
private void btn_Ok_Click(object sender, EventArgs e)
{
    Form_Kinematics.r = r; Form_Kinematics.R = R;
    Form_Kinematics.L = L; Form_Kinematics.H = H;
    this.Close();
}
}
}
```

ДОДАТОК Г. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ Form_Length

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Length : Form
    {
        private static double r_min, r_max, r;
        private static double R_min, R_max, R;
        private static double L_min, L_max, L;
        private static double H_min, H_max, H;
        private static double d, d_stat, d_max;

        public Form_Length()
        {
            InitializeComponent();

            r_min = 0.25; r_max = 0.75;
            r = MainForm.r;
            Bar_r1.Maximum = (int)(100 * r_max - 100 * r_min);
            Bar_r1.Value = (int)(Bar_r1.Maximum * (r - r_min) / (r_max
- r_min));
            lbl_r1.Text = string.Format("{0:F2} м", r);

            R_min = 1.00; R_max = 2.00;
            R = MainForm.R;
            Bar_r2.Maximum = (int)(100 * R_max - 100 * R_min);
            Bar_r2.Value = (int)(Bar_r2.Maximum * (R - R_min) / (R_max
- R_min));
            lbl_r2.Text = string.Format("{0:F2} м", R);

            L_min = 2.00; L_max = 5.00;
            L = MainForm.L;
            Bar_L.Maximum = (int)(100 * L_max - 100 * L_min);
            Bar_L.Value = (int)(Bar_L.Maximum * (L - L_min) / (L_max -
L_min));
            lbl_L.Text = string.Format("{0:F2} м", L);

            H_min = 3.00; H_max = 9.00;
            H = MainForm.H;
            Bar_H.Maximum = (int)(100 * H_max - 100 * H_min);
            Bar_H.Value = (int)(Bar_H.Maximum * (H - H_min) / (H_max -
H_min));
            lbl_H.Text = string.Format("{0:F2} м", H);

            d = MainForm.g * MainForm.m1 * MainForm.L / MainForm.C /
MainForm.Lc;
            d_stat = d * MainForm.r / MainForm.R;
            d_max = MainForm.nC * MainForm.Dc;

            if (d_stat > d_max)
            {
                label_alarm.Text = "Статична деформація пружини " +

```

```

        string.Format("{0:F4} (м)\r\n",
d_stat) +
        " перевищує допустиме значення "
+
        string.Format("{0:F4} (м) !",
d_max);
        label_alarm.ForeColor = Color.DarkRed;
    }
    else
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м).\r\n",
d_stat) +
            " Допустиме значення " +
            string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void Bar_r1_Scroll(object sender, EventArgs e)
{
    int m = Bar_r1.Value;
    r = r_min + m * (r_max - r_min) / Bar_r1.Maximum;
    lbl_r1.Text = string.Format("{0:F2} м", r);

    d_stat = d * r / R;

    if (d_stat > d_max)
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м)\r\n",
d_stat) +
            " перевищує допустиме значення "
+
            string.Format("{0:F4} (м) !",
d_max);
        label_alarm.ForeColor = Color.DarkRed;
    }
    else
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м).\r\n",
d_stat) +
            " Допустиме значення " +
            string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void Bar_r2_Scroll(object sender, EventArgs e)
{
    int m = Bar_r2.Value;
    R = R_min + m * (R_max - R_min) / Bar_r2.Maximum;
    lbl_r2.Text = string.Format("{0:F2} м", R);
}

```

```

        d_stat = d * r / R;

        if (d_stat > d_max)
        {
            label_alarm.Text = "Статична деформація пружини " +
                string.Format("{0:F4} (м)\r\n",
d_stat) +
                " перевищує допустиме значення "
+
                string.Format("{0:F4} (м) !",
d_max);
            label_alarm.ForeColor = Color.DarkRed;
        }
        else
        {
            label_alarm.Text = "Статична деформація пружини " +
                string.Format("{0:F4} (м).\r\n",
d_stat) +
                " Допустиме значення " +
                string.Format("{0:F4} (м).",
d_max);
            label_alarm.ForeColor = Color.DarkGreen;
        }
    }

    private void Bar_L_Scroll(object sender, EventArgs e)
    {
        int m = Bar_L.Value;
        L = L_min + m * (L_max - L_min) / Bar_L.Maximum;
        lbl_L.Text = string.Format("{0:F2} м", L);
    }

    private void Bar_H_Scroll(object sender, EventArgs e)
    {
        int m = Bar_H.Value;
        H = H_min + m * (H_max - H_min) / Bar_H.Maximum;
        lbl_H.Text = string.Format("{0:F2} м", H);
    }

    private void btn_Exit_Click(object sender, EventArgs e)
    {
        MainForm.clean = false; this.Close();
    }

    private void btn_Ok_Click(object sender, EventArgs e)
    {
        MainForm.r = r; MainForm.R = R; MainForm.L = L; MainForm.H
= H;
        MainForm.clean = true; this.Close();
    }
}
}

```

ДОДАТОК Д. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ Form_Kinematics

```

using System;
using System.Drawing;
using System.Threading;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Kinematics : Form
    {
        private Form_Length_K f_length_K;

        public static double r, R, L, H;
        public static double R2, L2;
        public static double Phi_Max, Psi_Max, Psi_Min, Phi_ext;

        public const double grad = 180.0 / Math.PI, dpi = 2 * Math.PI,
eps = 1.0E-8;
        public const float fgrad = (float)grad;
        public const int N2 = 360, M2 = N2 + 1, N3 = 180, M3 = N3 + 1;

        public static C1.Win.C1Chart.ChartDataSeries sAB;
        public static C1.Win.C1Chart.ChartDataSeries s2;
        public static C1.Win.C1Chart.ChartDataSeries s3;
        public static C1.Win.C1Chart.ChartDataSeries sL;
        public static C1.Win.C1Chart.ChartDataSeries sR;
        public static C1.Win.C1Chart.ChartDataSeries sf;
        public static C1.Win.C1Chart.ChartDataSeries sa;
        public static C1.Win.C1Chart.ChartDataSeries si;

        public Form_Kinematics()
        {
            InitializeComponent();
            r = MainForm.r;
            R = MainForm.R;
            L = MainForm.L;
            H = MainForm.H;
            Parameters_K();
        }

        // Функция, вычисляющая вспомогательные параметры задачи

        public void Parameters_K()
        {
            double alfa, O2O3, phi, psi, x_max, y_max, x_min, y_min,
dx, dy;
            int M_max, N_max; string txt = "";

            btn_Graf.Enabled = true; tBx_table.Text = txt;

            label_r1.Text = " r = " + string.Format("{0:F2}", r);
            label_R2.Text = " R = " + string.Format("{0:F2}", R);
            label_L.Text = " L = " + string.Format("{0:F2}", L);
            label_H.Text = " H = " + string.Format("{0:F2}", H);

            R2 = R * R; L2 = L * L;

```

```

// Определяем предельно допустимые значения для углов
O2O3 = Math.Sqrt(H * H + (L - R) * (L - R));
alfa = Math.Acos((L - R) / O2O3);
Psi_Max = Math.Acos((L * L - R * (L + H)) / L / O2O3) -
alfa;
Psi_Min = Math.Acos((L * L - R * (L - H)) / L / O2O3) -
alfa;

if (double.IsNaN(Psi_Max)) btn_Graf.Enabled = false;
if (double.IsNaN(Psi_Min)) btn_Graf.Enabled = false;

lbl_psi_max.Text = string.Format("psi_max = {0:F2}", grad
* Psi_Max);
lbl_psi_min.Text = string.Format("psi_min = {0:F2}", grad
* Psi_Min);

Phi_ext = 2.0 * Math.Atan(H / R);
Phi_Max = Math.Acos((R * (L - H) - H * H - R * R) / (H +
R) / O2O3) - alfa;

if (double.IsNaN(Phi_Max)) btn_Graf.Enabled = false;

lbl_phi_max.Text = string.Format("phi_max = {0:F2}", grad
* Phi_Max);
lbl_phi_ext.Text = string.Format("phi_* = {0:F2}", grad *
Phi_ext);

c1_Chart.ChartGroups[0].ChartData.SeriesList.Clear();

// Масштабируем изображение

x_max = R + H + L;
x_min = R + H - L; if (x_min >= 0) x_min = 0;
y_max = Math.Max(2 * R, L); y_min = 0;
M_max = c1_Chart.ChartArea.Size.Width;
N_max = c1_Chart.ChartArea.Size.Height;

dx = (x_max - x_min) / M_max; dy = (y_max - y_min) /
N_max;

if (dx > dy)
{
    c1_Chart.ChartArea.AxisX.Max = x_max;
    c1_Chart.ChartArea.AxisY.Max = N_max * dx;
}
else
{
    c1_Chart.ChartArea.AxisX.Max = M_max * dy;
    c1_Chart.ChartArea.AxisY.Max = y_max;
}

// Изображаем первичный рисунок

```

```

s2 = new C1.Win.C1Chart.ChartDataSeries();
c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s2);
s2.LineStyle.Color = Color.DarkBlue;
s2.LineStyle.Thickness = 2;
s2.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
s2.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
s2.Label = null;

PointF[] data2 =
(PointF[])Array.CreateInstance(typeof(PointF), M2);
for (int i = 0; i <= N2; i++)
{
    phi = i / grad;

    data2[i].X = (float)(R * (1.0 + Math.Sin(phi)));
    data2[i].Y = (float)(R * (1.0 - Math.Cos(phi)));
    txt += string.Format("{0}", i).PadLeft(6) +
        string.Format("{0:F5}", data2[i].X).PadLeft(11)
+
        string.Format("{0:F5}", data2[i].Y).PadLeft(11)
+ "\r\n";
}
txt += "\r\n\r\n";
s2.PointData.CopyDataIn(data2);

// Рисуем траекторию стержня 3

s3 = new C1.Win.C1Chart.ChartDataSeries();
c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(s3);
s3.LineStyle.Color = Color.DarkGreen;
s3.LineStyle.Thickness = 2;
s3.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
s3.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
s3.Label = null;

PointF[] data3 =
(PointF[])Array.CreateInstance(typeof(PointF), M3);

for (int i = -N3 / 2; i <= N3 / 2; i++)
{
    psi = i / grad; int j = i + N3 / 2;

    data3[j].X = (float)(R + H + L * Math.Sin(psi));
    data3[j].Y = (float)(L * (1.0 - Math.Cos(psi)));
    txt += string.Format("{0}", i).PadLeft(6) +
        string.Format("{0:F5}", data3[j].X).PadLeft(11)
+
        string.Format("{0:F5}", data3[j].Y).PadLeft(11)
+ "\r\n";
}
s3.PointData.CopyDataIn(data3);

```

```

// Рисуем стержень AB

sAB = new C1.Win.C1Chart.ChartDataSeries();
c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sAB);
sAB.LineStyle.Color = Color.Black;
sAB.LineStyle.Thickness = 3;
sAB.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.Dot;
sAB.SymbolStyle.Size = 10;
sAB.SymbolStyle.Color = Color.Black;

sAB.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sAB.Label = null;

PointF[] dataAB =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
dataAB[0].X = (float)R; dataAB[1].X = (float)(R + H);
dataAB[0].Y = 0; dataAB[1].Y = 0;
sAB.PointData.CopyDataIn(dataAB);

// Рисуем радиус тела 2

sR = new C1.Win.C1Chart.ChartDataSeries();
c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sR);
sR.LineStyle.Color = Color.Blue;
sR.LineStyle.Thickness = 1;
sR.SymbolStyle.Shape = C1.Win.C1Chart.SymbolShapeEnum.Dot;
sR.SymbolStyle.Size = 5;
sR.SymbolStyle.Color = Color.Blue;

sR.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sR.Label = null;

PointF[] dataR =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
dataR[0].X = (float)R; dataR[1].X = (float)R;
dataR[0].Y = (float)R; dataR[1].Y = 0;
sR.PointData.CopyDataIn(dataR);

// Рисуем стержень 3

sL = new C1.Win.C1Chart.ChartDataSeries();
c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sL);
sL.LineStyle.Color = Color.Green;
sL.LineStyle.Thickness = 1;
sL.SymbolStyle.Shape = C1.Win.C1Chart.SymbolShapeEnum.Dot;
sL.SymbolStyle.Size = 5;
sL.SymbolStyle.Color = Color.Green;

sL.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sL.Label = null;

PointF[] dataL =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
dataL[0].X = (float)(R + H); dataL[1].X = (float)(R + H);
dataL[0].Y = (float)L; dataL[1].Y = 0;

```

```

        sL.PointData.CopyDataIn(dataL);
    }

    // Функция, вычисляющая значение угла Psi (2) New

    public static double PSI(double phi)
    {
        double a, b, c, fa, fc, fb; int k;

        if (L == R) return phi;

        if (phi == 0) return 0;
        k = 0;
        if ((0 < phi) && (phi <= Phi_ext))
        {
            a = -eps; b = Psi_Max; fa = F_1(phi, a);
            do
            {
                c = (a + b) * 0.5; fc = F_1(phi, c); k++;
                if ((fa * fc) < 0) b = c; else a = c;
                if ((b - a) < eps) break;
            }
            while (Math.Abs(fc) > eps);
        }
        else
        {
            a = Psi_Min; b = eps; fb = F_1(phi, b);
            do
            {
                c = (a + b) * 0.5; fc = F_1(phi, c); k++;
                if ((fb * fc) < 0) a = c; else b = c;
                if ((b - a) < eps) break;
            }
            while (Math.Abs(fc) > eps);
        }
        return c;
    }

    // Функция, реализующая вычисления по формуле (1) или (2)

    public static double F_1(double phi, double psi)
    {
        double x = R * (1.0 - Math.Cos(phi)) - L * (1.0 -
Math.Cos(psi));
        double y = H + L * Math.Sin(psi) - R * Math.Sin(phi);
        return x * x + y * y - H * H;
    }

    private void btn_length_Click(object sender, EventArgs e)
    {
        f_length_K = new Form_Length_K();
        f_length_K.Activate();
        f_length_K.ShowDialog();
        Parameters_K();
    }

    private void btn_Graf_Click(object sender, EventArgs e)

```

```

    {
        double xA, xB, yA, yB, phi, psi, psi_old = 0, psi_new,
d_psi = 0;
        double ABx, ABo = H, err = 100.0 * eps;
        string txt = "", str = "error"; bool flag;
        btn_Graf.Enabled = false; btn_length.Enabled = false;
        Application.DoEvents();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Remove(sR);

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Remove(sL);
        tBx_table.Text = "";
        lbl_ABo.Text = string.Format("AB = {0:F6} м", ABo);
        lbl_ABo.Refresh();

        for (int i = 1; i <= 360; i++)
        {

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Remove(sAB);
            phi = i / grad;
            if (phi > dpi) phi = phi - dpi;

            xA = R * (1.0 + Math.Sin(phi));
            yA = R * (1.0 - Math.Cos(phi));

psi_old;

            psi = PSI(phi); psi_new = psi; d_psi = psi_new -
psi_old;

            xB = R + H + L * Math.Sin(psi);
            yB = L * (1.0 - Math.Cos(psi));

            // Контроль за длиной стержня AB

            ABx = Math.Sqrt((xB - xA) * (xB - xA) + (yB - yA) *
(yB - yA));
            flag = Math.Abs(ABo - ABx) > err;
            if (flag)
                lbl_ABx.ForeColor = Color.DarkRed;
            else
            {
                lbl_ABx.ForeColor = Color.DarkGreen;
                str = string.Format("  ABx = {0:F6} м", ABx);
            }
            lbl_ABx.Text = string.Format("AB = {0:F6} м", ABx);
            lbl_ABx.Refresh();

            // Рисуем стержень AB
            sAB = new C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sAB);
            sAB.LineStyle.Color = Color.Black;
            sAB.LineStyle.Thickness = 3;
            sAB.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.Dot;
            sAB.SymbolStyle.Size = 10;
            sAB.SymbolStyle.Color = Color.Black;

```

```

sAB.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sAB.Label = null;

PointF[] dataAB =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
dataAB[0].X = (float)xA; dataAB[1].X = (float)xB;
dataAB[0].Y = (float)yA; dataAB[1].Y = (float)yB;
sAB.PointData.CopyDataIn(dataAB);

// Рисуем радиус тела 2

sR = new C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sR);
sR.LineStyle.Color = Color.Gray;
sR.LineStyle.Thickness = 1;
sR.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.Dot;
sR.SymbolStyle.Size = 3;
sR.SymbolStyle.Color = Color.Blue;

sR.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sR.Label = null;

PointF[] dataR =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
dataR[0].X = (float)R; dataR[1].X = (float)xA;
dataR[0].Y = (float)R; dataR[1].Y = (float)yA;
sR.PointData.CopyDataIn(dataR);

// Рисуем стержень 3

sL = new C1.Win.C1Chart.ChartDataSeries();

c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sL);
sL.LineStyle.Thickness = 1;
sL.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.Dot;
sL.SymbolStyle.Size = 3;

if (d_psi > 0)
{
    sL.LineStyle.Color = Color.Green;
    sL.SymbolStyle.Color = Color.Green;
}
else
{
    sL.LineStyle.Color = Color.Orange;
    sL.SymbolStyle.Color = Color.Orange;
}

sL.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
sL.Label = null;

PointF[] dataL =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

```

```

        dataL[0].X = (float)(R + H); dataL[1].X = (float)xB;
        dataL[0].Y = (float)L; dataL[1].Y = (float)yB;
        sL.PointData.CopyDataIn(dataL);

        txt += string.Format("{0}", i).PadLeft(6) +
        string.Format("{0:F5}", phi * grad).PadLeft(11) +
        string.Format("{0:F5}", psi * grad).PadLeft(11);

        if (flag) txt += str + "\r\n"; else txt += "\r\n";

        Thread.Sleep(20);
        c1_Chart.Refresh();
        psi_old = psi_new;
    }

    // Рисуем радиус максимального угла Phi_Max

    sf = new C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sf);
    sf.LineStyle.Color = Color.Maroon;
    sf.LineStyle.Thickness = 2;
    sf.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sf.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
    sf.Label = null;

    PointF[] dataf =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
    dataf[0].X = (float)R; dataf[1].X = (float)(R * (1.0 +
Math.Tan(Phi_Max)));
    dataf[0].Y = (float)R; dataf[1].Y = 0;
    sf.PointData.CopyDataIn(dataf);

    // Рисуем радиус максимального угла Psi_Max

    sa = new C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(sa);
    sa.LineStyle.Color = Color.Orange;
    sa.LineStyle.Thickness = 2;
    sa.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
    sa.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
    sa.Label = null;

    PointF[] dataa =
(PointF[])Array.CreateInstance(typeof(PointF), 2);
    dataa[0].X = (float)(R + H); dataa[1].X = (float)(R + H +
L * Math.Tan(Psi_Max));
    dataa[0].Y = (float)L; dataa[1].Y = 0;
    sa.PointData.CopyDataIn(dataa);

    // Рисуем радиус минимального угла Psi_Min

    si = new C1.Win.C1Chart.ChartDataSeries();
    c1_Chart.ChartGroups.Group0.ChartData.SeriesList.Add(si);
    si.LineStyle.Color = Color.Green;
    si.LineStyle.Thickness = 2;

```

```

        si.SymbolStyle.Shape =
C1.Win.C1Chart.SymbolShapeEnum.None;
        si.FitType = C1.Win.C1Chart.FitTypeEnum.Line;
        si.Label = null;

        PointF[] datai =
(PointF[])Array.CreateInstance(typeof(PointF), 2);

        double yy, xx = R + H + L * Math.Tan(Psi_Min);

        if (xx < 0) { yy = Math.Abs(xx) / Math.Tan(-Psi_Min); xx =
0; }
        else yy = 0;

        datai[0].X = (float)(R + H); datai[1].X = (float)xx;

        datai[0].Y = (float)L; datai[1].Y = (float)yy;
        si.PointData.CopyDataIn(datai);

        Application.DoEvents();
        btn_Graf.Enabled = true; btn_length.Enabled = true;
        tBx_table.Text = txt;
    }
}
}

```

ДОДАТОК Е. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ `Form_Init`

```

using System;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Init : Form
    {
        private static double ksi_min, ksi_max, ksi_init, phi;

        public Form_Init()
        {
            InitializeComponent();

            ksi_min = -MainForm.Ksi_Max; ksi_max = MainForm.Ksi_Max;
ksi_init = MainForm.Ksi_Ini;
            Bar_ksi.Maximum = (int)(100 * ksi_max - 100 * ksi_min);
            Bar_ksi.Value = (int)(Bar_ksi.Maximum * (ksi_init -
ksi_min) / (ksi_max - ksi_min));
            lbl_ksi.Text = string.Format("{0:F2} м", ksi_init);
            phi = (ksi_init / MainForm.r) * MainForm.grad;
            lbl_phi_o.Text = "Початковий кут відхилення тіла 2 = "
                + string.Format("{0:F0} гр", phi);
        }

        private void Bar_ksi_Scroll(object sender, EventArgs e)
        {
            int m = Bar_ksi.Value;
            ksi_init = ksi_min + m * (ksi_max - ksi_min) /
Bar_ksi.Maximum;
            lbl_ksi.Text = string.Format("{0:F2} м", ksi_init);

            phi = (ksi_init / MainForm.r) * MainForm.grad;
            lbl_phi_o.Text = "Початковий кут відхилення тіла 2 = "
                + string.Format("{0:F0} гр", phi);
        }

        private void btn_Exit_Click(object sender, EventArgs e)
        {
            MainForm.clean = false; this.Close();
        }

        private void btn_Ok_Click(object sender, EventArgs e)
        {
            MainForm.clean = true; MainForm.Ksi_Ini = ksi_init;
this.Close();
        }
    }
}

```

ДОДАТОК Є. ЛІСТИНГ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ КЛАСУ Form_Coil

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace Diplom_Voskolup_fin
{
    public partial class Form_Coil : Form
    {
        private static double C_min, C_max, C;
        private static double dc_min, dc_max, d_coil;
        private static double n3_min, n3_max, n3;
        private static double d, d_stat, d_max;

        public Form_Coil()
        {
            InitializeComponent();

            C_min = 100.0; C_max = 500.0;
            C = MainForm.C;
            Bar_C.Maximum = (int)(1 * C_max - 1 * C_min);
            Bar_C.Value = (int)(Bar_C.Maximum * (C - C_min) / (C_max -
C_min));
            lbl_C.Text = string.Format("{0:F1} H/м", C);

            dc_min = 1.00; dc_max = 2.00;
            d_coil = MainForm.Dc;
            Bar_dc.Maximum = (int)(100 * dc_max - 100 * dc_min);
            Bar_dc.Value = (int)(Bar_dc.Maximum * (d_coil - dc_min) /
(dc_max - dc_min));
            lbl_dc.Text = string.Format("{0:F2} м", d_coil);

            n3_min = 0.60; n3_max = 0.90;
            n3 = MainForm.n3;
            Bar_n3.Maximum = (int)(100 * n3_max - 100 * n3_min);
            Bar_n3.Value = (int)(Bar_n3.Maximum * (n3 - n3_min) /
(n3_max - n3_min));
            lbl_n3.Text = string.Format("{0:F2}", n3);

            d = MainForm.m1 * MainForm.g * MainForm.r / MainForm.R;
            d_stat = d / C / n3;
            d_max = MainForm.nC * MainForm.Dc;

            if (d_stat > d_max)
            {
                label_alarm.Text = "Статична деформація пружини " +
string.Format("{0:F4} (м)\r\n",
d_stat) +
                " перевищує допустиме значення " +
                string.Format("{0:F4} (м) !",
d_max);
                label_alarm.ForeColor = Color.DarkRed;
            }
            else
            {

```

```

        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м).\r\n",
d_stat) +
                " Допустиме значення " +
                string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void Bar_C_Scroll(object sender, EventArgs e)
{
    int m = Bar_C.Value;
    C = C_min + m * (C_max - C_min) / Bar_C.Maximum;
    lbl_C.Text = string.Format("{0:F1} Н/м", C);

    d_stat = d / C / n3; d_max = 0.15 * d_coil;

    if (d_stat > d_max)
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м)\r\n",
d_stat) +
                " перевищує допустиме значення "
+
                string.Format("{0:F4} (м) !",
d_max);
        label_alarm.ForeColor = Color.DarkRed;
    }
    else
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м).\r\n",
d_stat) +
                " Допустиме значення " +
                string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void Bar_dc_Scroll(object sender, EventArgs e)
{
    int m = Bar_dc.Value;
    d_coil = dc_min + m * (dc_max - dc_min) / Bar_dc.Maximum;
    lbl_dc.Text = string.Format("{0:F2} м", d_coil);

    d_stat = d / C / n3; d_max = 0.15 * d_coil;

    if (d_stat > d_max)
    {
        label_alarm.Text = "Статична деформація пружини " +
            string.Format("{0:F4} (м)\r\n",
d_stat) +
                " перевищує допустиме значення "
+

```

```

                                string.Format("{0:F4} (м) !",
d_max);
                                label_alarm.ForeColor = Color.DarkRed;
                                }
                                else
                                {
                                label_alarm.Text = "Статична деформація пружини " +
                                string.Format("{0:F4} (м).\r\n",
d_stat) +
                                " Допустиме значення " +
                                string.Format("{0:F4} (м).",
d_max);
                                label_alarm.ForeColor = Color.DarkGreen;
                                }
                                }

private void Bar_n3_Scroll(object sender, EventArgs e)
{
    int m = Bar_n3.Value;
    n3 = n3_min + m * (n3_max - n3_min) / Bar_n3.Maximum;
    lbl_n3.Text = string.Format("{0:F2}", n3);

    d_stat = d / C / n3; d_max = 0.15 * d_coil;

    if (d_stat > d_max)
    {
        label_alarm.Text = "Статична деформація пружини " +
        string.Format("{0:F4} (м)\r\n",
d_stat) +
        " перевищує допустиме значення "
+
        string.Format("{0:F4} (м) !",
d_max);
        label_alarm.ForeColor = Color.DarkRed;
    }
    else
    {
        label_alarm.Text = "Статична деформація пружини " +
        string.Format("{0:F4} (м).\r\n",
d_stat) +
        " Допустиме значення " +
        string.Format("{0:F4} (м).",
d_max);
        label_alarm.ForeColor = Color.DarkGreen;
    }
}

private void btn_Exit_Click(object sender, EventArgs e)
{
    MainForm.clean = false; this.Close();
}
private void btn_Ok_Click(object sender, EventArgs e)
{
    MainForm.C = C; MainForm.Dc = d_coil; MainForm.n3 = n3;
    MainForm.clean = true; this.Close();
}
}}

```