

Міністерство освіти і науки України
Сумський державний університет
Кафедра комп'ютерних наук
Секція інформаційно-комунікаційних технологій

Пояснювальна записка
до курсової роботи
з дисципліни
«Програмування»

Викладач	Прокопенко В. М.
Студент	Крячко А.К
Група	КН-41/2
Варіант	5

Зміст

Постановка задачі.....	3
Теоретичний матеріал із теми	5
Опис структури даних та вимоги для них	7
Алгоритм роботи програми	9
Опис функцій користувача	12
Опис файлів та їх призначення	15
Список використаних бібліотек	17
Інструкція для роботи з програмою	19
Приклад тестування та результати роботи програми	20
Графіки	22
Висновки	23
Список літератури	24

Постановка задачі

Варіант 5

Описати масив структур із трьох елементів. Кожна структура об'єднує дані для одного варіанта розрахунку. Необхідно для кожного варіанта на відрізку часу від 0 до T із кроком Δt побудувати графік зміни активного струму () а I A через конденсатор:

$$I_a = U \omega c t_g \delta ,$$

де U – напруга (В); ω – колова частота (рад/с); c – ємність конденсатора (ф); $\text{tg} \delta$ – тангенс кута діелектричних втрат.

Напруга і частота змінюються з часом:

$$U = \begin{cases} U_0(1 - e^{-kt}) \text{ для } t \in [0, T/2], \\ U_0(1 - e^{-kT/2})e^{-k(t-T/2)} \text{ для } t \in [T/2, T]; \end{cases}$$
$$k = \begin{cases} k_0 e^{mt} \text{ для } t \in [0, T/4], \\ k_0 e^{\frac{mT}{4}} \text{ для } t \in [\frac{T}{4}, \frac{3}{4}T], \\ k_0 e^{\frac{mT}{4}} e^{-m(t-\frac{3}{4}T)} \text{ для } t \in [\frac{3}{4}T, T]; \end{cases}$$
$$\omega = \begin{cases} \omega_0(1 + nt) \text{ для } t \in [0, \frac{T}{4}], \\ \omega_0(1 + n\frac{T}{4}) \text{ для } t \in [\frac{T}{4}, \frac{3T}{4}], \\ \omega_0(1 + n\frac{T}{4} - n(t - \frac{3T}{4})) \text{ для } t \in [\frac{3T}{4}, T], \end{cases}$$

де ω_0, k_0, m, U_0, n – задані константи.

Вхідні дані зчитуються з файла. Результати обчислень занести до іншого файла. Передбачити окремі функції для обчислення ω, k, U

Вхідні дані:

$$1. \quad T = 600 \text{ с}, \quad \Delta t = 30 \text{ с}, \quad U_0 = 200 \text{ В}, \quad k_0 = 0,001, \quad m = 0,001, \\ \omega_0 = 6,28 \cdot 10^6 \text{ рад/с}, \quad n = 0,001, \quad c = 100 \cdot 10^{-12} \text{ ф}, \quad t_g \delta = 10^{-3} .$$

2. $U_0 = 300 \text{ В}$, $k_0 = 0,0015$, $\omega_0 = 3,14 \cdot 10^6 \text{ рад/с}$, $c = 200 \cdot 10^{-12} \text{ ф}$.

Решта даних – див. пункт 1.

3. $U_0 = 250 \text{ В}$, $k_0 = 0,0012$, $\omega_0 = 12 \cdot 10^6 \text{ рад/с}$, $c = 400 \cdot 10^{-12} \text{ ф}$.

Решта даних – див. пункт 1.

Теоретичний матеріал із теми

- **Огляд мови програмування C та її застосування у розробленій програмі**
- Мова програмування C є фундаментальною у світі розробки, відома своєю давньою історією та значним впливом, що сприяв формуванню багатьох сучасних мов програмування. Вона належить до процедурного парадигми та вирізняється можливістю низькорівневого доступу до пам'яті, що забезпечує високоефективне управління системними ресурсами. Завдяки цим характеристикам C активно застосовується у створенні операційних систем, драйверів пристроїв, вбудованих систем та іншого програмного забезпечення, де критичною є висока продуктивність.
- У процесі розробки програми були використані наступні ключові програмні елементи:
- **Структури:** Це складні типи даних, які дозволяють групувати змінні різних типів під єдиним логічним іменем.
- **Функції:** Представляють собою відокремлені блоки коду, призначені для виконання конкретних завдань, і можуть бути викликані з будь-якої частини програми. У даній програмі функції `k_func`, `omega_func`, `U_func` та `calculate_and_save` відповідають за обчислення проміжних значень та активної потужності, тоді як функція `write_results` використовується для запису отриманих результатів у файл.
- **Масиви:** Це колекції елементів одного типу, що зберігаються у суміжних ділянках пам'яті. У ході виконання завдання були використані масиви `time` та `pa` для зберігання відповідних значень часу та потужності на кожному кроці розрахунку.
- **Константи:** Це змінні, значення яких залишається незмінним протягом всього виконання програми. Для спрощення обчислень були визначені константи `PI`

- Програма також активно використовує низку стандартних функцій:
- **Файлові операції (fopen(), fscanf(), fprintf(), fclose()):** Ці функції забезпечують взаємодію з файловою системою, дозволяючи відкривати файли, зчитувати та записувати дані, а також закривати файлові потоки. Вони були використані для читання вхідних даних з файлу input.c та збереження результатів у файл results.txt.
- **Математична функція exp():** Призначена для обчислення експоненціальної функції.
- **Виведення на екран printf():** Використовується для відображення інформації на консолі.
- **Системна команда System("chcp 65001"):** Ця функція виконує команду операційної системи, що дозволяє коректно відображати символи кирилиці.

Опис структури даних та вимоги для них

Таблиця 1 – Опис даних

Ім'я параметра у формулі	Змінна у програмі	Тип змінної	Призначення
t (час)	t	double	Поточне значення часу, яке ітерується в циклі для розрахунків.
T (період)	$d \rightarrow T$	double	Загальний період часу для розрахунку.
ω_0 (початкова кутова частота)	$d \rightarrow \omega_0$	double	Початкове значення кутової частоти для розрахунків.
n (коефіцієнт зміни частоти)	$d \rightarrow n$	double	Коефіцієнт, що впливає на зміну кутової частоти.
U_0 (початкова напруга)	$d \rightarrow U_0$	double	Початкове значення напруги для розрахунків.
c (ємність/інший коефіцієнт)	$d \rightarrow c$	double	Коефіцієнт, що використовується для обчислення струму I_a
$\tan(\delta)$ (тангенс кута втрат)	$d \rightarrow \tan_{\delta}$	double	Тангенс кута втрат, використовується для обчислення струму I_a
dt (крок часу)	$d \rightarrow dt$	double	Крок приросту часу в циклі розрахунку.
$k(t)$ (коефіцієнт k у момент часу t)	k	double	Розраховане значення коефіцієнта k на основі часу t та вхідних даних.
$\omega(t)$ (кутова частота у момент часу t)	ω	double	Розраховане значення кутової частоти на основі

			часу t та вхідних даних.
$U(t)$ (напруга у момент часу t)	U	double	Розраховане значення напруги на основі часу t та вхідних даних.
I_a (активний струм)	I_a	double	Розраховане значення активного струму
π (число Пі)	M_PI	Double(Макрос)	Математична константа π

Алгоритм роботи програми

АЛГОРИТМ РОБОТИ ПРОГРАМИ (MAIN.C)

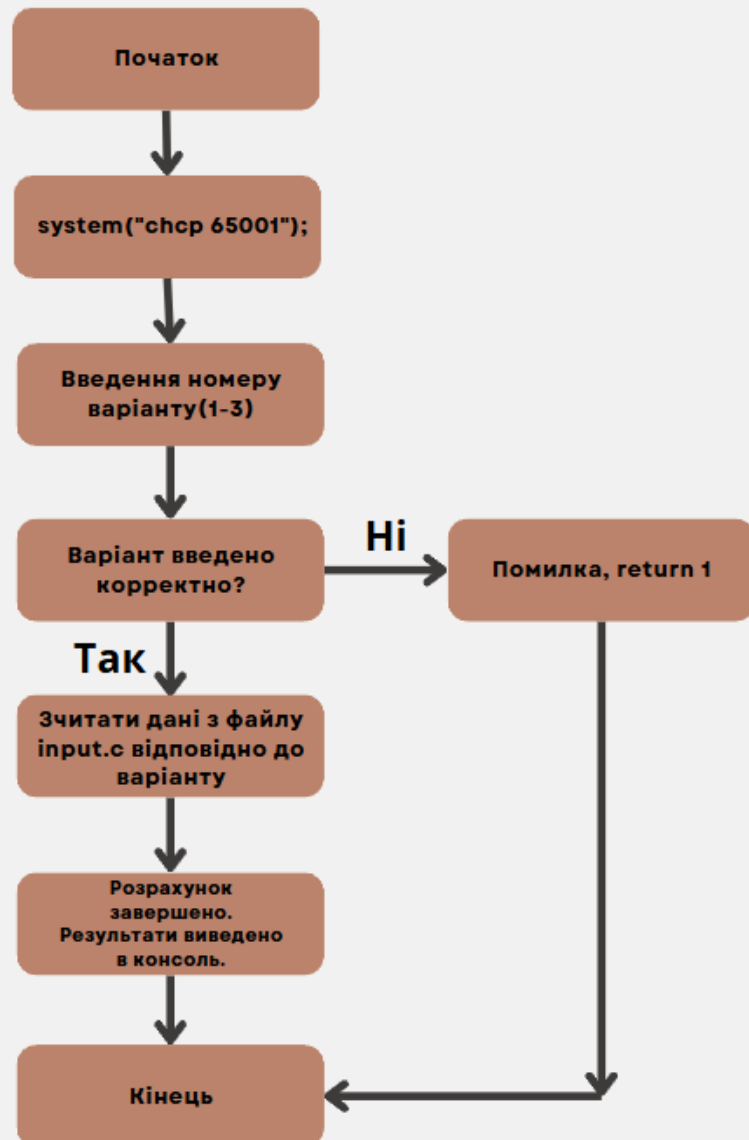


Рисунок 1 – Блок-схема main.c()

АЛГОРИТМ РОБОТИ ПРОГРАМИ (CALC.C, K_FUNC)

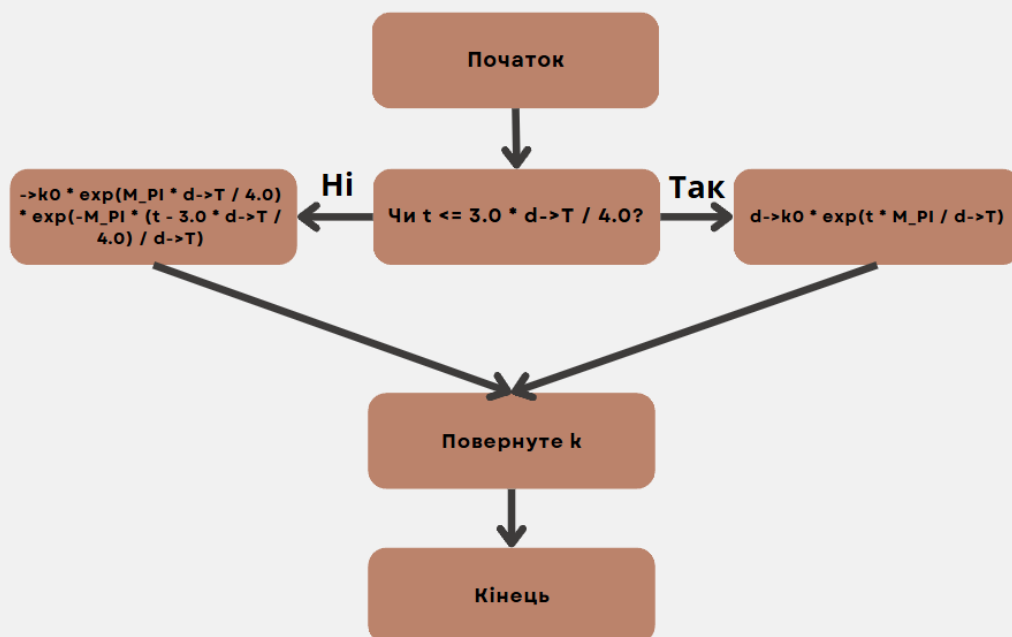


Рисунок 2 – Блок-схема k_func

АЛГОРИТМ РОБОТИ ПРОГРАМИ (CALC.C, OMEGA_FUNC)

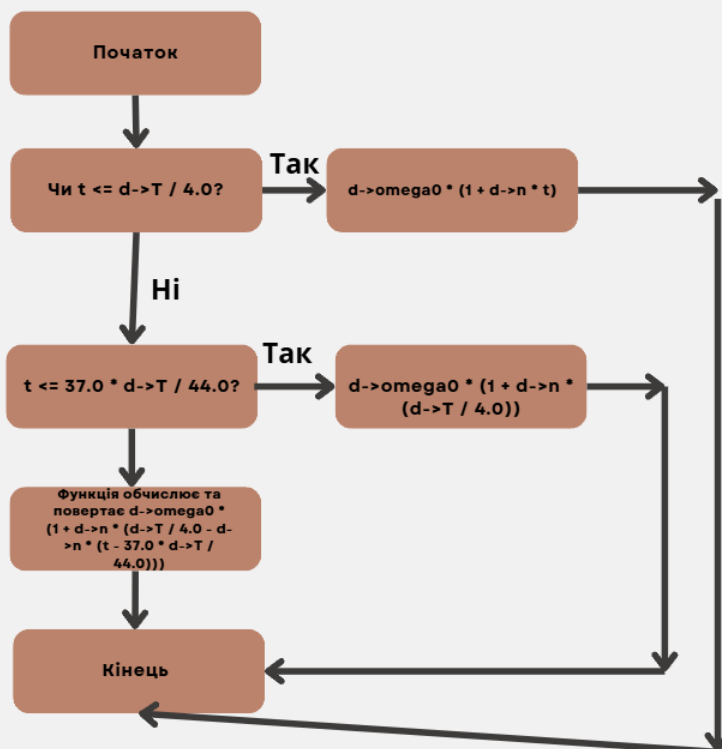


Рисунок 3 – Блок-схема omega_func

АЛГОРИТМ РОБОТИ ПРОГРАМИ (CALC.C, U_FUNC)

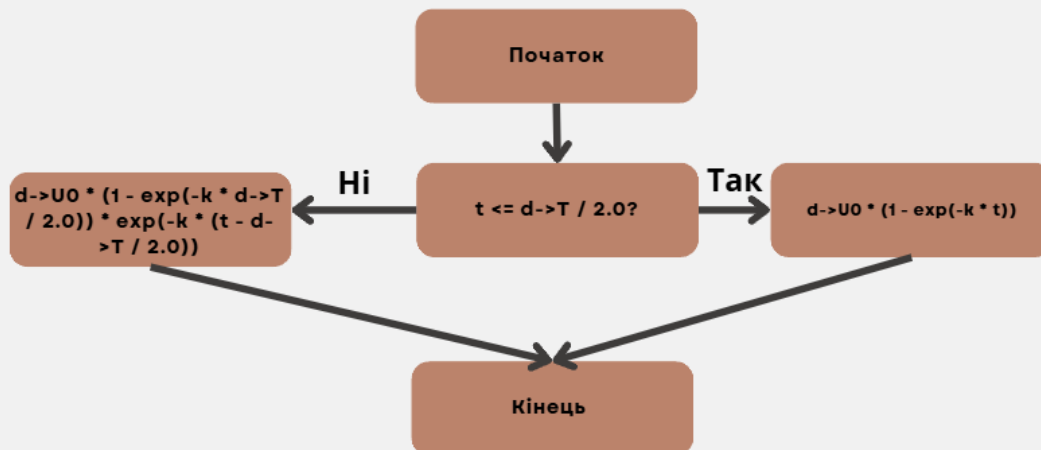


Рисунок 4 – Блок-схема u_func

АЛГОРИТМ РОБОТИ ПРОГРАМИ (CALC.C, CALCULATE_AND_SAVE)

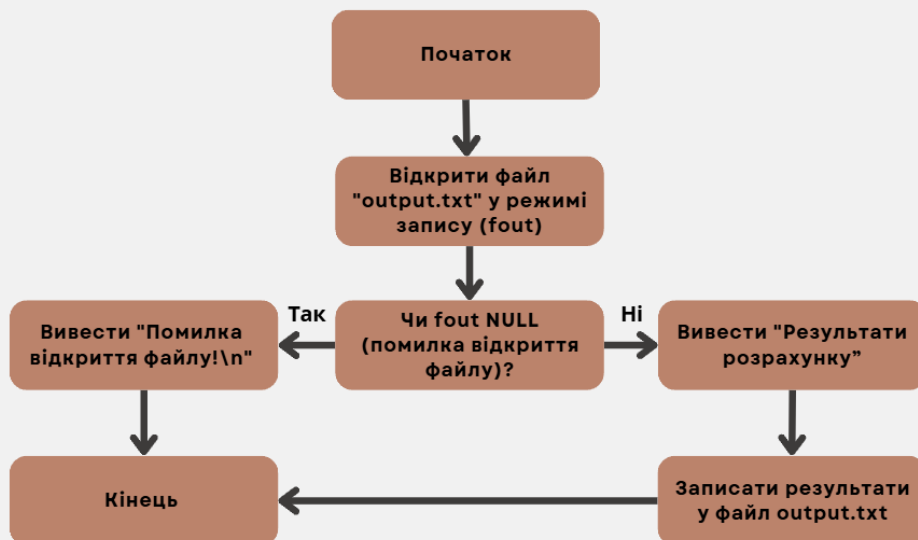


Рисунок 5 – Блок-схема calc_and_save

Опис функцій користувача

У процесі розробки програми було створено 4 функції, які реалізують логіку розрахунку активного струму I_a та збереження результатів.

1. Функція `static double k_func(double t, const InputData* d)`

- **Призначення:** Функція призначена для обчислення коефіцієнта k залежно від часу t . Цей коефіцієнт використовується у формулі для розрахунку напруги U .
- **Вхідні параметри:**
 - t (double): Поточний момент часу (с), для якого виконується розрахунок.
 - d (const InputData*): Вказівник на константну структуру, яка містить загальний час дослідження T та початкове значення коефіцієнта k_0 .
- **Вихід:**
 - Функція повертає значення k типу double, яке використовується для подальших обчислень у функції `U_func()`.

2. Функція `static double omega_func(double t, const InputData* d)`

- **Призначення:** Ця функція обчислює значення кутової частоти ω у певний момент часу t . Залежність ω від часу є кусково-заданою.
- **Вхідні параметри:**
 - t (double): Поточний момент часу (с), для якого обчислюється кутова частота ω .
 - d (const InputData*): Вказівник на константну структуру InputData, що містить загальний час T , початкову кутову частоту ω_0 та коефіцієнт n .
- **Вихід:**
 - Функція повертає значення кутової частоти ω типу double, яке передається до функції `calculate_and_save()` для розрахунку активного струму I_a .

3. Функція `static double U_func(double t, const InputData* d)`

- **Призначення:** Обчислює миттєву напругу U на основі часу t та змінного коефіцієнта k . Логіка функції є кусковою та залежить від попередньо обчисленого k .
- **Вхідні параметри:**
 - t (double): Поточний момент часу (с).
 - d (const InputData*): Вказівник на константну структуру InputData, що містить загальний час T та початкове значення напруги U_0 .
- **Логіка роботи:**
 - Функція спочатку викликає `k_func(t, d)` для отримання значення k .
 - На першому інтервалі (коли $t \leq d \rightarrow T / 2.0$), напруга розраховується за формулою $d \rightarrow U_0 * (1 - \exp(-k * t))$.
 - В іншому випадку (коли $t > d \rightarrow T / 2.0$), напруга обчислюється як $d \rightarrow U_0 * (1 - \exp(-k * d \rightarrow T / 2.0)) * \exp(-k * (t - d \rightarrow T / 2.0))$.
- **Вихід:**
 - Функція повертає значення напруги U типу double у відповідний момент часу, яке використовується для розрахунку активного струму I_a у функції `calculate_and_save()`.

4. Функція `void calculate_and_save(const InputData* d)`

- **Призначення:** Ця функція є головною для виконання всіх розрахунків та збереження результатів у файл "output.txt". Вона не повертає жодного значення (void).
- **Вхідні параметри:**
 - d (const InputData*): Вказівник на константну структуру InputData, що містить усі необхідні вхідні дані для розрахунків, включаючи $d \rightarrow T$, $d \rightarrow dt$, $d \rightarrow c$ та $d \rightarrow tg_delta$.
- **Логіка роботи:**
 - Функція намагається відкрити файл "output.txt" у режимі запису.

- Якщо файл не може бути відкрито, виводиться повідомлення про помилку на консоль, і функція завершується.
- У разі успішного відкриття, функція виводить заголовки результатів на консоль та у файл.
- Далі, вона ітерує змінну часу t від 0 до $d \rightarrow T$ з кроком $d \rightarrow dt$.
- У кожній ітерації циклу обчислюються U за допомогою $U_func()$, ω за допомогою $\omega_func()$, а потім активний струм I_a за формулою $U * \omega * d \rightarrow c * d \rightarrow tg_delta$.
- Обчислені значення t та I_a виводяться на консоль та записуються у файл "output.txt".
- Після завершення всіх розрахунків та запису, файл "output.txt" закривається.

Опис файлів та їх призначення

Для реалізації програми було задіяно 3 файли.

1. main.c

Файл main.c: Цей файл містить основну логіку програми, функцію main(), з якої починається виконання. Він відповідає за:

- Встановлення кодування для коректного відображення кирилиці (system("chcp 65001");).
- Запит у користувача номера варіанту розрахунку (від 1 до 3).
- Завантаження вхідних даних відповідно до обраного варіанту за допомогою функції load_input().
- Обробку помилок, якщо введено невірний номер варіанту.
- Виклик функції calculate_and_save() для виконання розрахунків та збереження результатів.
- Повідомлення користувача про завершення розрахунку.

2. input.h

Це заголовний файл, який містить оголошення структури InputData

Структура InputData об'єднує всі необхідні вхідні параметри для розрахунків, такі як T, dt, U0, k0, m, omega0, n, c, tg_delta.

Також у цьому файлі оголошено прототип функції load_input(), яка відповідає за завантаження даних для різних варіантів.

3. Input.c

Цей файл містить реалізацію функції load_input()

Вона приймає вказівник на структуру InputData та номер варіанту.

Використовуючи оператор switch, функція присвоює відповідні значення параметрам структури InputData залежно від вибраного варіанту (1, 2 або 3).

Повертає 1 у разі успішного завантаження даних і 0, якщо номер варіанту недійсний

4. Calc.h

Цей заголовний файл містить оголошення функції calculate_and_save().

Він включає `input.h`, що дозволяє використовувати структуру `InputData`.

Основна мета цього файлу – надати прототип функції, щоб інші файли могли її викликати.

5. `calc.c`

Цей файл містить реалізації функцій, що виконують математичні розрахунки, та функцію для збереження результатів.

- **`k_func()`**: Обчислює коефіцієнт k залежно від часу t та вхідних даних.
- **`omega_func()`**: Обчислює кутову частоту ω залежно від часу t та вхідних даних.
- **`U_func()`**: Обчислює напругу U залежно від часу t та обчисленого коефіцієнта k .
- **`calculate_and_save()`**: Ця функція відкриває файл `"output.txt"`, виконує цикл за часом, обчислює U , ω та активний струм I_a на кожному кроці, виводить їх на консоль та записує у файл. Вона також обробляє можливі помилки відкриття файлу.

6. `output.txt`

Це текстовий файл, який генерується програмою під час виконання.

Він містить результати розрахунків для кожного обраного варіанту.

Дані представлені у вигляді таблиці, де для кожного моменту часу t вказується відповідне значення активного струму

7. `Makefile`

Це текстовий файл, який автоматизує процес компіляції програми з вихідних файлів. Він містить набір правил та інструкцій для утиліти `make`, яка визначає, які файли потрібно перекомпілювати, щоб отримати кінцевий виконуваний файл

Список використаних бібліотек

1. <stdio.h>

Бібліотека <stdio.h> є однією з найважливіших і найчастіше використовуваних у мові програмування C. Вона відповідає за введення та виведення інформації, як на екран, так і в файли.

Функції, що використовуються в програмі:

- `fopen()` / `fclose()` - відповідають за відкриття та закриття файлів.
- `fscanf()` - використовується для зчитування вхідних параметрів з текстового файлу. Особливістю є те, що вона дозволяє форматовувати

вихідні дані, що критично важливо для коректної ініціалізації параметрів розрахунків.

- `fprintf()` - забезпечує запис результатів у вихідний файл у чітко структурованому вигляді.
- `printf()` - виводить проміжні результати розрахунків у консоль, що значно полегшує процес налагодження програми.

2. `<math.h>`

Бібліотека `<math.h>` надає потужний набір математичних функцій, які є невід'ємною частиною наукових та інженерних розрахунків. У нашому випадку вона використовується для реалізації складних обчислень, що описують фізичні процеси.

У програмі застосована лише одна функція із цієї бібліотеки:

- `exp()` - обчислює експоненційну функцію, яка відіграє головну роль у формулах для розрахунку напруги та частоти. Ця функція дозволяє точно моделювати процеси затухання та зростання.

3. `<stdlib.h>`

Бібліотека `<stdlib.h>` містить набір утиліт для роботи з операційною системою та виконання базових операцій [3].

Застосування в програмі:

- `system()` - дозволяє виконувати системні команди. У нашому випадку використовується для зміни кодування консолі на UTF-8 (сhcr 65001), що забезпечує коректне відображення кирилических символів.

Також у програмі використовуються макроси для визначення ключових констант:

```
#define M_PI 3.14159265358979323846
```

Інструкція для роботи з програмою

1. Підготовка до запуску Переконайтесь, що всі вихідні файли програми(main.c, input.c, calc.c, input.h, calc.h) знаходяться у одній папці.
2. Запуск програми Запустіть програму kurs.exe (Windows), або в консолі введіть ./kurs (Linux/MacOS)
3. Введення даних Введіть число 1,2 або 3 відповідно до бажаного варіанту і натисніть Enter. Якщо введено некоректне значення, програма завершить роботу з помилкою.
4. Отримання результатів Після введення коректного варіанту, програма обчислить струм за часом для заданих параметрів. Результати одночасно виведуться у консолі та збережуться у файл output.txt

Приклад тестування та результати роботи програми

Вхідні дані одразу для трьох варіантів

Вхідні дані:

1. $T = 600\text{с}$, $\Delta t = 30\text{с}$, $U_0 = 200\text{В}$, $k_0 = 0,001$, $m = 0,001$,
 $\omega_0 = 6,28 \cdot 10^6\text{рад/с}$, $n = 0,001$, $c = 100 \cdot 10^{-12}\text{ф}$, $t_{\sigma}\delta = 10^{-3}$.

2. $U_0 = 300\text{В}$, $k_0 = 0,0015$, $\omega_0 = 3,14 \cdot 10^6\text{рад/с}$, $c = 200 \cdot 10^{-12}\text{ф}$.

Решта даних – див. пункт 1.

3. $U_0 = 250\text{В}$, $k_0 = 0,0012$, $\omega_0 = 12 \cdot 10^6\text{рад/с}$, $c = 400 \cdot 10^{-12}\text{ф}$.

Решта даних – див. пункт 1.

Нижче наведені відповідні результати роботи програми (Рисунок 6, Рисунок 7, Рисунок 8)

```
Введіть номер варіанту (1-3): 1
Результати розрахунку:
t (с), Ia (A)
t = 0.00 с, Ia = 0.00000000e+000 A
t = 30.00 с, Ia = 4.46238243e-006 A
t = 60.00 с, Ia = 1.04994999e-005 A
t = 90.00 с, Ia = 1.83816013e-005 A
t = 120.00 с, Ia = 2.83357946e-005 A
t = 150.00 с, Ia = 4.04939319e-005 A
t = 180.00 с, Ia = 5.34340977e-005 A
t = 210.00 с, Ia = 6.75582539e-005 A
t = 240.00 с, Ia = 8.22867166e-005 A
t = 270.00 с, Ia = 9.68397379e-005 A
t = 300.00 с, Ia = 1.10325552e-004 A
t = 330.00 с, Ia = 9.94554761e-005 A
t = 360.00 с, Ia = 8.38013162e-005 A
t = 390.00 с, Ia = 6.50376629e-005 A
t = 420.00 с, Ia = 4.56780303e-005 A
t = 450.00 с, Ia = 2.84211647e-005 A
t = 480.00 с, Ia = 0.00000000e+000 A
t = 510.00 с, Ia = 0.00000000e+000 A
t = 540.00 с, Ia = 0.00000000e+000 A
t = 570.00 с, Ia = 0.00000000e+000 A
t = 600.00 с, Ia = 0.00000000e+000 A
```

Рисунок 6 – Результат роботи програми (1 частина)

```

Введіть номер варіанту (1-3): 2
Результати розрахунку:
t (с), Ia (A)
t = 0.00 с, Ia = 0.00000000e+000 A
t = 30.00 с, Ia = 9.95327384e-006 A
t = 60.00 с, Ia = 2.31518022e-005 A
t = 90.00 с, Ia = 3.99375911e-005 A
t = 120.00 с, Ia = 6.04280832e-005 A
t = 150.00 с, Ia = 8.43905284e-005 A
t = 180.00 с, Ia = 1.08304276e-004 A
t = 210.00 с, Ia = 1.32523900e-004 A
t = 240.00 с, Ia = 1.55503358e-004 A
t = 270.00 с, Ia = 1.75671546e-004 A
t = 300.00 с, Ia = 1.91791188e-004 A
t = 330.00 с, Ia = 1.54822415e-004 A
t = 360.00 с, Ia = 1.13587933e-004 A
t = 390.00 с, Ia = 7.41656041e-005 A
t = 420.00 с, Ia = 4.20064664e-005 A
t = 450.00 с, Ia = 1.99995156e-005 A
t = 480.00 с, Ia = 0.00000000e+000 A
t = 510.00 с, Ia = 0.00000000e+000 A
t = 540.00 с, Ia = 0.00000000e+000 A
t = 570.00 с, Ia = 0.00000000e+000 A
t = 600.00 с, Ia = 0.00000000e+000 A

```

Рисунок 7 – Результат роботи програми (2 частина)

```

Введіть номер варіанту (1-3): 3
Результати розрахунку:
t (с), Ia (A)
t = 0.00 с, Ia = 0.00000000e+000 A
t = 30.00 с, Ia = 5.14332994e-005 A
t = 60.00 с, Ia = 1.20461159e-004 A
t = 90.00 с, Ia = 2.09642576e-004 A
t = 120.00 с, Ia = 3.20746047e-004 A
t = 150.00 с, Ia = 4.54102792e-004 A
t = 180.00 с, Ia = 5.92434922e-004 A
t = 210.00 с, Ia = 7.38967072e-004 A
t = 240.00 с, Ia = 8.86096889e-004 A
t = 270.00 с, Ia = 1.02473450e-003 A
t = 300.00 с, Ia = 1.14581335e-003 A
t = 330.00 с, Ia = 9.86976059e-004 A
t = 360.00 с, Ia = 7.85558648e-004 A
t = 390.00 с, Ia = 5.67890244e-004 A
t = 420.00 с, Ia = 3.65212309e-004 A
t = 450.00 с, Ia = 2.03738715e-004 A
t = 480.00 с, Ia = 0.00000000e+000 A
t = 510.00 с, Ia = 0.00000000e+000 A
t = 540.00 с, Ia = 0.00000000e+000 A
t = 570.00 с, Ia = 0.00000000e+000 A
t = 600.00 с, Ia = 0.00000000e+000 A

```

Рисунок 8 – Результат роботи програми (3 частина)

Графіки

Графіки

Для візуалізації результатів роботи програми було побудовано графіки залежності активного струму I_a від часу t для кожного з трьох варіантів розрахунків.

Враховуючи експоненційний характер зміни параметрів та малі значення струму, для побудови графіків було обрано напівлогарифмічний масштаб, що дозволило наочно відобразити динаміку процесу.

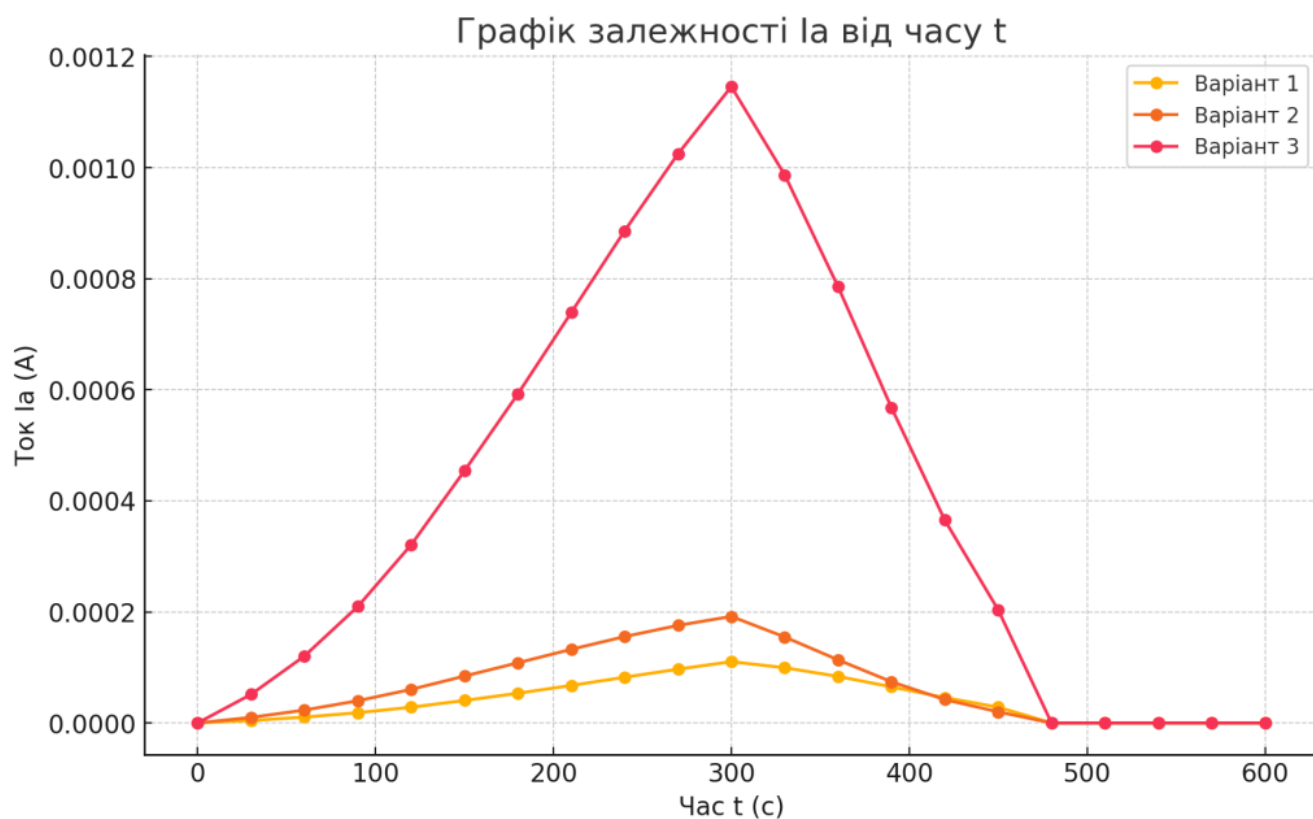


Рисунок 9 – Графік результатів програми(1-3 варіант)

Висновки

Висновки

Програма успішно реалізує обчислення активного струму I_a залежно від часу, ґрунтуючись на змінах напруги U , коефіцієнта k та кутової частоти ω , які також змінюються з часом за кусково-заданими функціями.

Основні досягнення та результати:

- **Модульна архітектура:** Програма демонструє ефективне використання модульного підходу, розділяючи логіку на окремі функції (`k_func`, `omega_func`, `U_func`, `calculate_and_save`). Це значно покращує читабельність, підтримуваність та можливість повторного використання коду.
- **Гнучкість вхідних даних:** Завдяки використанню структури `InputData` та функції `load_input`, програма може легко адаптуватися до різних наборів вхідних параметрів, що дозволяє проводити розрахунки для кількох варіантів без зміни основного коду.
- **Ефективна робота з файлами:** Реалізовано зчитування вхідних даних з файлу (через логіку завантаження варіантів) та збереження результатів у файл `"output.txt"`, що забезпечує постійне зберігання даних та зручність аналізу.
- **Математична точність:** Використання типу `double` для всіх розрахунків та математичної бібліотеки (`<math.h>`) забезпечує необхідну точність обчислень для експоненційних залежностей та малих значень параметрів.
- **Візуалізація даних:** Наявність згенерованих результатів в `output.txt` створює основу для подальшої візуалізації (наприклад, побудови графіків залежності I_a від t), що дозволяє наочно аналізувати динаміку процесу для кожного варіанту. Враховуючи характер змін, напівлогарифмічний масштаб графіків є доцільним для відображення динаміки малих значень.

Загалом, розроблена програма є надійним інструментом для моделювання та аналізу заданих фізичних процесів, що демонструє ключові аспекти процедурного програмування на C та ефективної обробки даних.

Список літератури

1. Авраменко В. В. Програмування [Електронний ресурс] : навчальний курс / Віктор Васильович Авраменко. – Режим доступу: <https://mix.sumdu.edu.ua/textbooks/104505/index.html>. – Назва з екрана.
2. Громов Ю.Ю., Татарніков О.М. Електричні та магнітні властивості діелектриків: навч. посібник. – К. : Наукова думка, 2018. – 232 с.
3. Керніган Б., Рітчі Д. Керівництво користувача мови С. – К. : Вільямс. – 2002. – 272 с.
4. Методичні вказівки до курсової роботи з дисципліни «Програмування» / уклад.: Авраменко В. В., Боровик В. О., Тиркусова Н. В. – Суми : Сумський державний університет, 2021. – 43 с.