

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

Горовий К. С.

Група

КН-41/2

Варіант

6

Зміст

Постановка задачі.....	3
Теоретичний матеріал із теми	5
Опис структури даних та вимоги для них	7
Алгоритм роботи програми	9
Опис функцій користувача	13
Опис файлів та їх призначення	16
Список використаних бібліотек	17
Інструкція для роботи з програмою	19
Приклад тестування та результати роботи програми	21
Графіки	23
Висновки	24
Список літератури	25

Постановка задачі

Варіант 6

Описати масив структур із 3-х елементів. Кожна структура об'єднує дані для одного варіанту розрахунку. Необхідно для кожного варіанту на відрізьку часу від 0 до T із кроком Δt побудувати графік зміни активної потужності P_a (Вт), що розсіюється в діелектрику.

$$P_a = u^2 2\pi f c \cdot \operatorname{tg}(\delta),$$

де u – діюче значення змінної напруги (В);

f – частота змінного струму (Гц);

c – ємність конденсатора (Ф);

$\operatorname{tg}(\delta)$ – тангенс кута діелектричних втрат.

Напруга u і частота f змінюються у часі:

$$u = \begin{cases} u_0(1 - e^{-kt}) \text{ для } t \in [0, \frac{T}{4}] \\ u_0(1 - e^{-\frac{kT}{4}}) \text{ для } t \in [\frac{T}{4}, \frac{T}{2}] \\ u_0(1 - e^{-kt})e^{-k(t - \frac{T}{2})} \text{ для } t \in [\frac{T}{2}, T] \end{cases}$$

$$k = \begin{cases} k_0(1 + mt) \text{ для } t \in [0, \frac{T}{2}] \\ k_0(1 + m\frac{T}{2}) \text{ для } t \in [\frac{T}{2}, T] \end{cases}$$

Тут u_0, k_0, m – задані константи.

$$f = \begin{cases} f_0(1 + ht) \text{ для } t \in [0, \frac{T}{4}] \\ f_0(1 + h\frac{T}{4}) \text{ для } t \in [\frac{T}{4}, \frac{3T}{4}] \\ f_0(1 + h\frac{T}{4}) - f_0 e^{-h(t - \frac{3T}{4})} \text{ для } t \in [\frac{3T}{4}, T] \end{cases}$$

f_0, h – задані константи.

Вхідні дані зчитуються з файлу. Результати обчислень занести в інший файл. Передбачити окремі функції для обчислення f , k і u .

Вхідні дані:

1. $T = 10$ с, $\Delta t = 0,5$ с, $u_0 = 100$ В, $k_0 = 0,1$, $m = 0,1$, $f_0 = 1000$ Гц, $h = 0,1$, $C = 1000 \cdot 10^{-12}$ ф, $tg(\delta) = 10^{-3}$.

2. $u_0 = 120$ В, $m = 0,15$, $f_0 = 1500$ Гц, $h = 0,12$, $C = 2000 \cdot 10^{-12}$ ф, $tg(\delta) = 0,005$. Решту даних див. пункт 1.

3. $u_0 = 90$ В, $m = 0,2$, $f_0 = 2000$ Гц, $h = 0,15$, $C = 4000 \cdot 10^{-12}$ ф, $tg(\delta) = 0,008$. Решту даних див. пункт 1.

Теоретичний матеріал із теми

Мова програмування C є однією з найстаріших і водночас найвпливовіших, адже вона стала основою для багатьох сучасних мов програмування. Це процедурна мова, яка має низькорівневий доступ до пам'яті, що дозволяє ефективно керувати ресурсами комп'ютера. Її часто використовують для розробки операційних систем, драйверів, вбудованих систем та іншого програмного забезпечення, де потрібна висока продуктивність [3].

Основні визначення, що були використані в процесі розроблення програми:

- Структура - це складений тип даних, який дозволяє об'єднувати різні типи змінних під одним ім'ям. У програмі використовується структура `VariantData` для зберігання вхідних параметрів кожного варіанту розрахунку.
- Функція - блок коду, який виконує певну задачу та може бути викликаний з інших частин програми. У програмі функції `calculate_k`, `calculate_u`, `calculate_f`, `calculate_pa` використовуються для обчислення проміжних значень та активної потужності, а `write_results` – для заповнення файлу з результатами.
- Масив - набір елементів одного типу, які зберігаються в послідовних комірках пам'яті. При виконанні завдання були використані масиви `time` та `pa` для зберігання значень часу та потужності на кожному кроці.
- Константа - змінна, значення якої не може змінюватися під час виконання програми. Використовував `PI`, `NUM_VARIANTS`, `MAX_TIME_STEPS` для зручності в обчисленнях.

Стандартні функції, що використовувалися:

- `Fopen()`, `fscanf()`, `fprintf()`, `fclose()` – для відкриття файлу, зчитування і запису даних, закриття файлу. Використовувалися для зчитування вхідних даних з файлу `input.txt` та запису результатів у файл `results.txt`.
- `Exp()` - математична функція для обчислення експоненти.
- `Printf()` - виводить дані на екран.

- `System("chcp 65001")` - виконує системну команду для коректного відображення кирилиці.

Опис структури даних та вимоги для них

Таблиця 1 – Опис даних

Ім'я параметра у формулі	Змінна у програмі	Тип змінної	Призначення
T	T	double	Загальний час розрахунку (с), вхідний параметр
Δt	dt	double	Крок часу (с), вхідний параметр
u0	u0	double	Початкове значення напруги (В), вхідний параметр
k0	k0	double	Коефіцієнт для розрахунку k, вхідний параметр
m	m	double	Коефіцієнт для розрахунку m, вхідний параметр
f0	f0	double	Початкова частота (Гц), вхідний параметр
h	h	double	Коефіцієнт для розрахунку частоти f, вхідний параметр
c	c	double	Ємність конденсатора (ф), вхідний параметр
$\text{tg}(\delta)$	tg_delta	double	Тангенс кута діелектричних втрат, вхідний параметр
t	t	double	Поточний час (с), вихідне значення
k	k	double	Коефіцієнт, що залежить від часу, вихідне значення
u	u	double	Діюче значення напруги (В), вихідне значення

f	f	double	Частота змінного струму (Гц), вихідне значення
P _a	pa[i]	double	Активна потужність, що розсіюється в діелектрику (Вт), вихідне значення

Алгоритм роботи програми

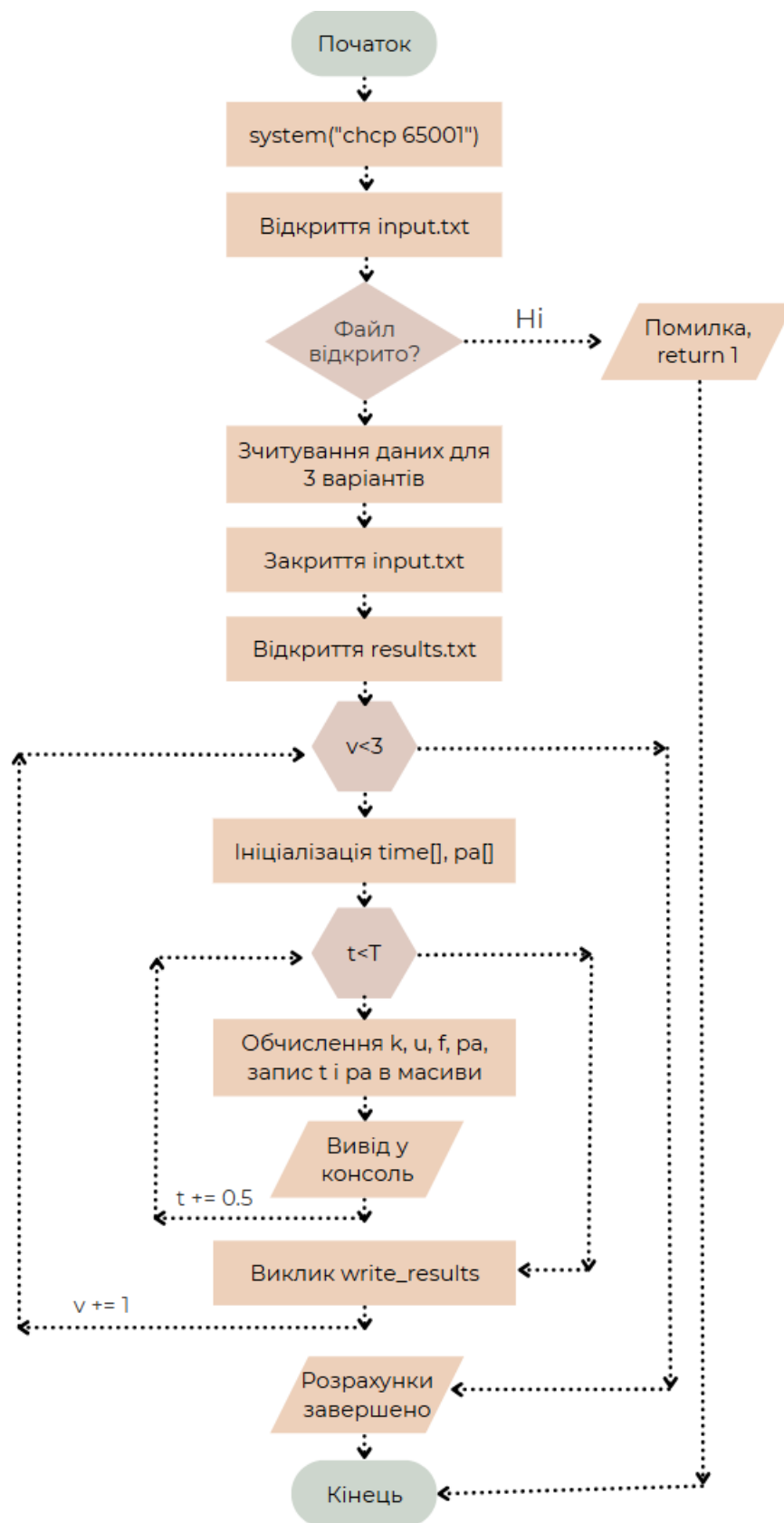


Рисунок 1 – Блок-схема `main()`

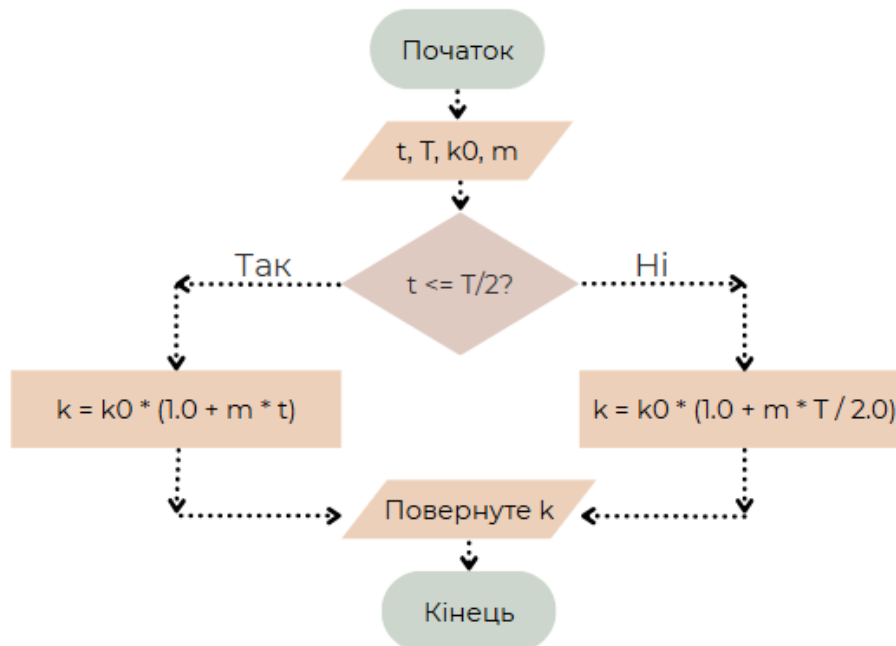


Рисунок 2 – Блок-схема `calculate_k()`

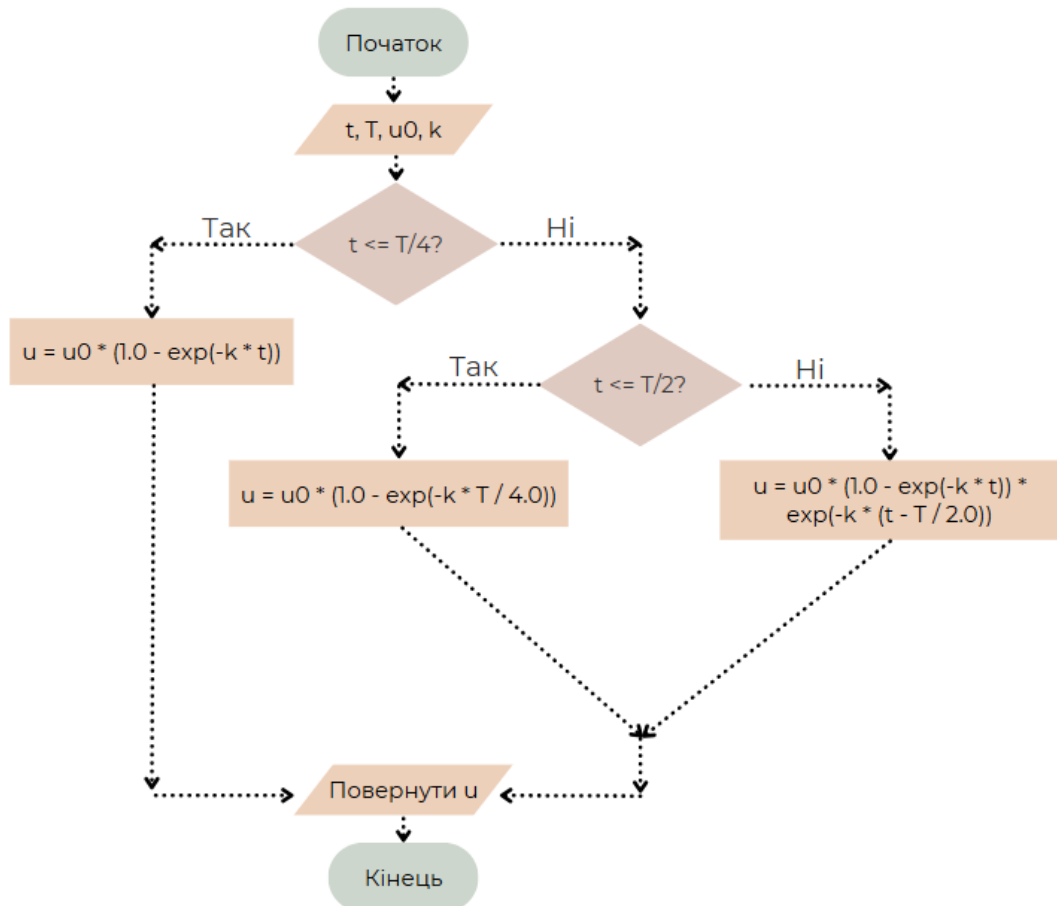


Рисунок 3 – Блок-схема `calculate_u()`

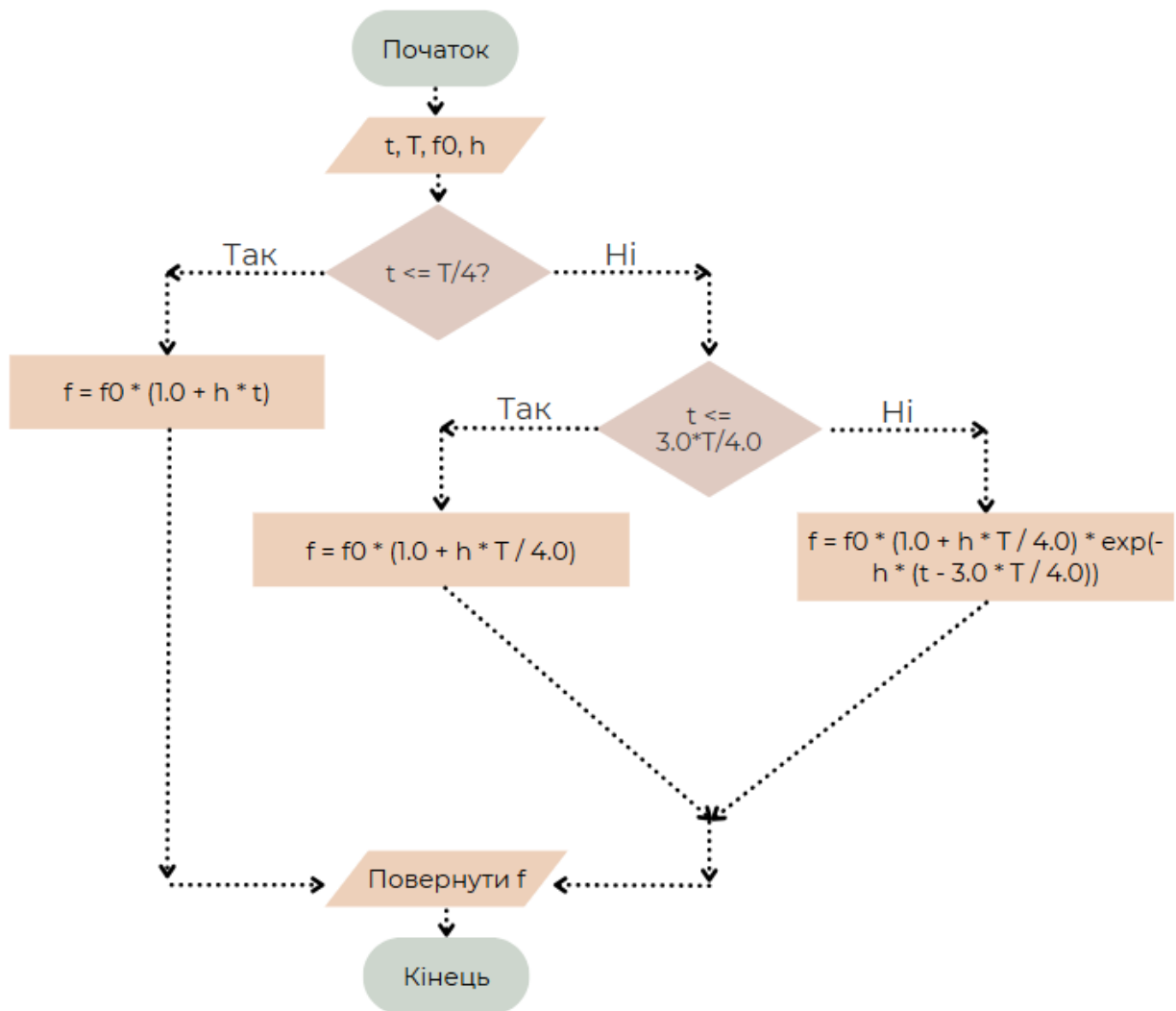


Рисунок 4 – Блок-схема `calculate_f()`

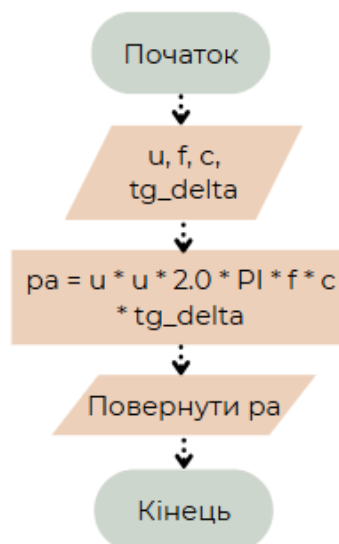


Рисунок 5 – Блок-схема `calculate_pa()`

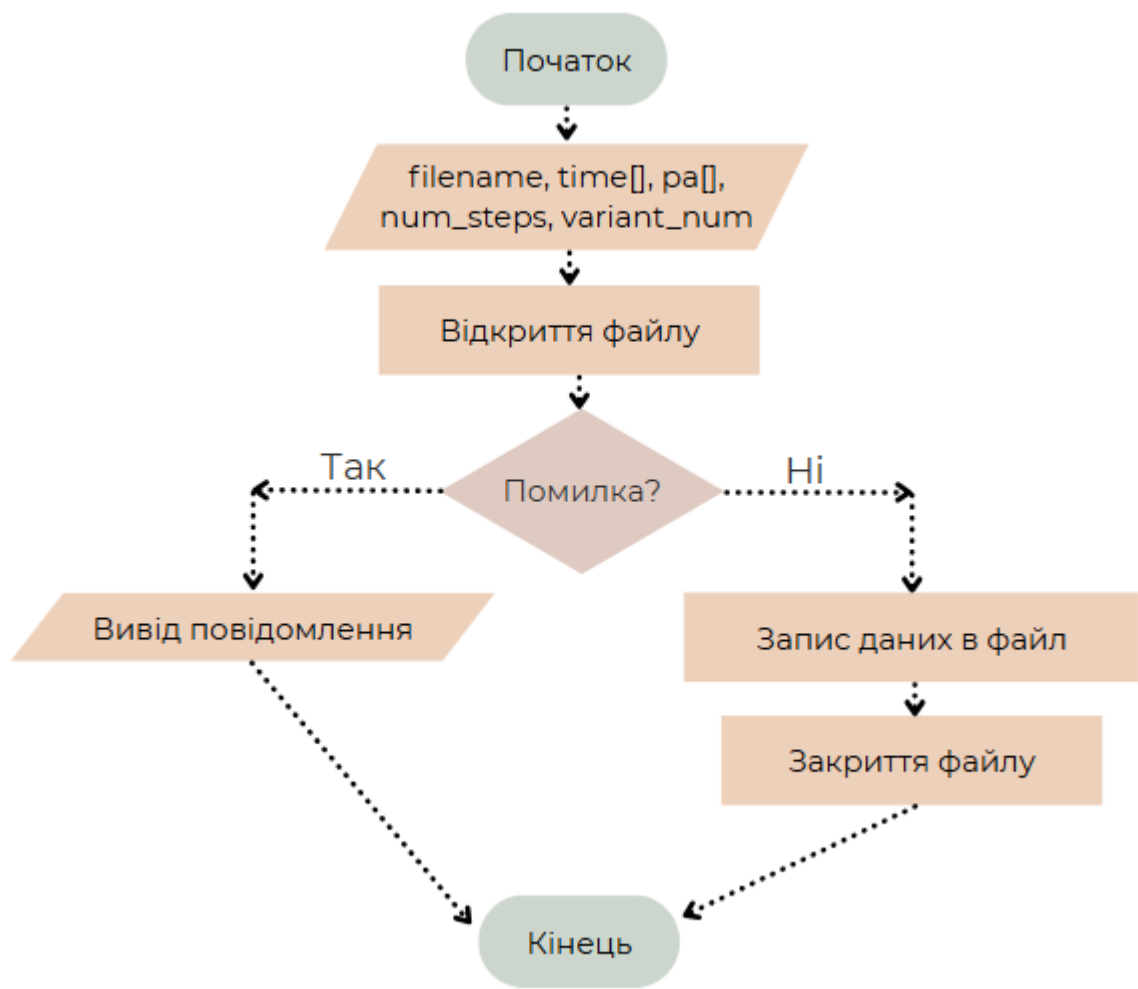


Рисунок 6 – Блок-схема write_results()

Опис функцій користувача

У процесі розробки програми було створено 5 функцій, які реалізують логіку розрахунку активної потужності.

1. Функція `calculate_k()`

Функція призначена для обчислення коефіцієнта жорсткості залежно від часу, використовується у формулі для розрахунку напруги [2].

Вхідні параметри:

- `t (double)` - поточний момент часу (с), для якого виконується розрахунок.
- `T (double)` - загальний час дослідження (с).
- `k0 (double)` - початкове значення коефіцієнта.
- `m (double)` - коефіцієнт, що визначає швидкість зростання `k` у часі.

Вихід:

Функція повертає значення `k` типу `double`, яке використовується для подальших обчислень у функції `calculate_u()`.

2. Функція `calculate_u()`

Обчислює миттєву напругу на основі часу та змінного коефіцієнта `k`, з урахуванням трьох фаз зміни напруги (розгін, стабілізація, спад).

Вхідні параметри:

- `t (double)` - поточний момент часу (с).
- `T (double)` - загальний час дослідження (с).
- `u0 (double)` - початкове значення напруги (В).
- `k (double)` - коефіцієнт, розрахований функцією `calculate_k()`.

Логіка роботи:

На першому інтервалі ($t \leq T/4$) напруга зростає, на другому ($T/4 < t \leq T/2$) - стабілізується на рівні, досягнутому на момент $T/4$, на третьому ($t > T/2$) - напруга починає спадати з урахуванням експоненційного затухання.

Вихід:

Повертає значення напруги u типу `double` у відповідний момент часу, використовується для розрахунку потужності у функції `calculate_pa()`.

3. Функція `calculate_f()`

Функція обчислює частоту змінного струму, яка змінюється у часі залежно від параметрів.

Вхідні параметри:

- t (`double`) - поточний момент часу (с).
- T (`double`) - загальний час дослідження (с).
- f_0 (`double`) - початкова частота (Гц).
- h (`double`) - коефіцієнт, що визначає швидкість зміни частоти.

Логіка роботи:

На першому інтервалі ($t \leq T/4$) частота лінійно зростає, на другому ($T/4 < t \leq 3T/4$) - стабілізується на рівні, досягнутому на момент $T/4$, на третьому ($t > 3T/4$) - частота експоненційно спадає.

Вихід:

Функція повертає значення частоти f типу `double`, яке передається до `calculate_pa()`

4. Функція `calculate_pa()`

Функція обчислює активну потужність, що розсіюється в діелектрику конденсатора, на основі напруги, частоти, ємності та тангенса кута втрат.

Вхідні параметри:

- u (`double`) - миттєве значення напруги (В).
- f (`double`) - миттєва частота (Гц).
- c (`double`) - ємність конденсатора (Ф).
- tg_delta (`double`) - тангенс кута діелектричних втрат.

Вихід:

Функція повертає значення потужності p_a типу `double`.

5. Функція `write_results()`

Функція записує у файл результати розрахунків для кожного варіанту у вигляді таблиці.

Вхідні параметри:

- filename (const char*) - назва файлу для запису.
- time[] (double[]) - масив із значеннями часу.
- pa[] (double[]) - масив із значеннями потужності.
- num_steps (int) - кількість кроків часу.
- variant_num (int) - номер варіанту.

Вихід:

Нічого не повертає, лише записує дані у файл.

Опис файлів та їх призначення

Для реалізації програми було задіяно 3 файли.

1. main.c

Файл main.c є основним файлом програми, який реалізує всю логіку розрахунків та взаємодії з файлами:

- Оголошує структуру даних.
- Містить опис всіх функцій користувача.
- Реалізує функцію main(), що координує весь процес. Це включає зчитування даних з файлу input.txt, обчислення параметрів, запис результатів у файл results.txt.

2. input.txt

Текстовий файл для вхідних даних, використовується для ініціалізації параметрів моделювання.

Структура файла:

Файл містить три рядки, кожен з яких відповідає одному варіанту розрахунків. Дані у кожному рядку розділені пробілами та впорядковані у строгому порядку:

$T \Delta t u_0 k_0 m f_0 h c \operatorname{tg} \delta$

3. results.txt

Текстовий файл для вихідних даних, зберігає результати розрахунків для кожного варіанту у вигляді таблиці з часом t та активною потужністю P_a .

Структура файла:

Для кожного варіанту створюється окремий блок даних із заголовком. Дані записуються у форматі:

Час (с)	P_a (Вт)
t_1	p_{a1}
t_2	p_{a2}
...	

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 формувати вхідні 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 (схр 65001), що забезпечує коректне відображення кириличних символів.

Також у програмі використовуються макроси для визначення ключових констант:

```
#define PI 3.14159265358979323846
```

```
#define NUM_VARIANTS 3
```

```
#define MAX_TIME_STEPS 21
```

Інструкція для роботи з програмою

1. Необхідні ресурси

Для коректної роботи програми необхідно:

- Операційна система: Windows/Linux/macOS
- Компілятор C: GCC (GNU Compiler Collection) або інший сучасний компілятор
- Файл `input.txt`, що містить параметри для обчислення для кожного з трьох варіантів.

2. Підготовка даних

- Створити файл програми з кодом (`main.c`) і зберегти його у робочій папці.
- Створити файл `input.txt` у тій самій папці, де знаходиться код програми `main.c`, заповнити його даними за таким шаблоном (тільки числові значення, розділені пробілами):

`T Δt u0 k0 m f0 h c tgδ`

`T Δt u0 k0 m f0 h c tgδ`

`T Δt u0 k0 m f0 h c tgδ`

3. Компіляція і запуск

Відкрийте термінал/командний рядок у папці з файлом `main.c` і виконайте команду для компіляції:

```
gcc main.c -o program -lm
```

Після цього введіть команду для запуску:

```
./program
```

4. Взаємодія з користувачем

Програма не вимагає введення даних вручну під час роботи, взаємодія повністю автоматизована. Всі дані беруться з підготовленого вхідного файлу, користувачу не потрібно відповідати на жодні запити. Після завершення розрахунків користувач отримає повідомлення про успішне збереження результатів.

Ось короткий алгоритм дій програми:

- Зчитує дані з input.txt.
- Виконує розрахунки для кожного варіанту.
- Виводить проміжні результати у консоль.
- Записує фінальні результати у файл results.txt.

5. Перевірка та обробка помилок

- Якщо файл input.txt не знайдено або пошкоджений - виводиться повідомлення «Err with file!».
- Якщо в одному з рядків не вказано потрібну кількість чисел - виводиться помилка «Помилка зчитування даних для варіанту X!»
- Якщо файл results.txt неможливо створити для запису - виводиться повідомлення про помилку.

Приклад тестування та результати роботи програми

Вхідні дані одразу для трьох варіантів (файл input.txt):

10.0 0.5 100.0 0.1 0.1 1000.0 0.1 1000.0e-12 1.0e-3

10.0 0.5 120.0 0.1 0.15 1500.0 0.12 2000.0e-12 0.005

10.0 0.5 90.0 0.1 0.2 2000.0 0.15 4000.0e-12 0.008

Нижче наведені відповідні результати роботи програми (Рисунок 7, Рисунок 8, Рисунок 9)

```
Обчислення для варіанту 1:  
t=0.00с: u=0.0000В, f=1000.0000Гц, Pa=0.000000000000Вт  
t=0.50с: u=5.1146В, f=1050.0000Гц, Pa=0.000000172579Вт  
t=1.00с: u=10.4166В, f=1100.0000Гц, Pa=0.000000749935Вт  
t=1.50с: u=15.8442В, f=1150.0000Гц, Pa=0.000001813914Вт  
t=2.00с: u=21.3372В, f=1200.0000Гц, Pa=0.000003432705Вт  
t=2.50с: u=26.8384В, f=1250.0000Гц, Pa=0.000005657236Вт  
t=3.00с: u=27.7473В, f=1250.0000Гц, Pa=0.000006046864Вт  
t=3.50с: u=28.6448В, f=1250.0000Гц, Pa=0.000006444386Вт  
t=4.00с: u=29.5312В, f=1250.0000Гц, Pa=0.000006849389Вт  
t=4.50с: u=30.4066В, f=1250.0000Гц, Pa=0.000007261473Вт  
t=5.00с: u=31.2711В, f=1250.0000Гц, Pa=0.000007680251Вт  
t=5.50с: u=52.1174В, f=1250.0000Гц, Pa=0.000021333154Вт  
t=6.00с: u=51.0770В, f=1250.0000Гц, Pa=0.000020489956Вт  
t=6.50с: u=49.7322В, f=1250.0000Гц, Pa=0.000019425188Вт  
t=7.00с: u=48.1578В, f=1250.0000Гц, Pa=0.000018214745Вт  
t=7.50с: u=46.4159В, f=1250.0000Гц, Pa=0.000016920908Вт  
t=8.00с: u=44.5578В, f=1189.0368Гц, Pa=0.000014832799Вт  
t=8.50с: u=42.6256В, f=1131.0468Гц, Pa=0.000012912265Вт  
t=9.00с: u=40.6538В, f=1075.8850Гц, Pa=0.000011172415Вт  
t=9.50с: u=38.6700В, f=1023.4134Гц, Pa=0.000009615665Вт  
t=10.00с: u=36.6967В, f=973.5010Гц, Pa=0.000008237038Вт
```

Рисунок 7 – Результат роботи програми (1 частина)

Обчислення для варіанту 2:

t=0.00с:	u=0.0000В,	f=1500.0000Гц,	Pa=0.000000000000Вт
t=0.50с:	u=6.2797В,	f=1590.0000Гц,	Pa=0.000003939650Вт
t=1.00с:	u=13.0361В,	f=1680.0000Гц,	Pa=0.000017938331Вт
t=1.50с:	u=20.1427В,	f=1770.0000Гц,	Pa=0.000045122207Вт
t=2.00с:	u=27.4738В,	f=1860.0000Гц,	Pa=0.000088212592Вт
t=2.50с:	u=34.9073В,	f=1950.0000Гц,	Pa=0.000149295238Вт
t=3.00с:	u=36.4879В,	f=1950.0000Гц,	Pa=0.000163121722Вт
t=3.50с:	u=38.0391В,	f=1950.0000Гц,	Pa=0.000177286635Вт
t=4.00с:	u=39.5616В,	f=1950.0000Гц,	Pa=0.000191761781Вт
t=4.50с:	u=41.0558В,	f=1950.0000Гц,	Pa=0.000206520294Вт
t=5.00с:	u=42.5222В,	f=1950.0000Гц,	Pa=0.000221536586Вт
t=5.50с:	u=67.9537В,	f=1950.0000Гц,	Pa=0.000565771596Вт
t=6.00с:	u=65.4839В,	f=1950.0000Гц,	Pa=0.000525392414Вт
t=6.50с:	u=62.7035В,	f=1950.0000Гц,	Pa=0.000481724167Вт
t=7.00с:	u=59.7217В,	f=1950.0000Гц,	Pa=0.000436996829Вт
t=7.50с:	u=56.6249В,	f=1950.0000Гц,	Pa=0.000392853062Вт
t=8.00с:	u=53.4816В,	f=1836.4408Гц,	Pa=0.000330038670Вт
t=8.50с:	u=50.3445В,	f=1729.4949Гц,	Pa=0.000275425245Вт
t=9.00с:	u=47.2546В,	f=1628.7769Гц,	Pa=0.000228522935Вт
t=9.50с:	u=44.2425В,	f=1533.9243Гц,	Pa=0.000188652477Вт
t=10.00с:	u=41.3307В,	f=1444.5955Гц,	Pa=0.000155049746Вт

Рисунок 8 – Результат роботи програми (2 частина)

Обчислення для варіанту 3:

t=0.00с:	u=0.0000В,	f=2000.0000Гц,	Pa=0.000000000000Вт
t=0.50с:	u=4.8163В,	f=2150.0000Гц,	Pa=0.000010027715Вт
t=1.00с:	u=10.1772В,	f=2300.0000Гц,	Pa=0.000047897291Вт
t=1.50с:	u=15.9449В,	f=2450.0000Гц,	Pa=0.000125238681Вт
t=2.00с:	u=21.9795В,	f=2600.0000Гц,	Pa=0.000252544178Вт
t=2.50с:	u=28.1440В,	f=2750.0000Гц,	Pa=0.000437958643Вт
t=3.00с:	u=29.6712В,	f=2750.0000Гц,	Pa=0.000486779904Вт
t=3.50с:	u=31.1607В,	f=2750.0000Гц,	Pa=0.000536880327Вт
t=4.00с:	u=32.6135В,	f=2750.0000Гц,	Pa=0.000588107153Вт
t=4.50с:	u=34.0303В,	f=2750.0000Гц,	Pa=0.000640317310Вт
t=5.00с:	u=35.4122В,	f=2750.0000Гц,	Pa=0.000693376876Вт
t=5.50с:	u=54.3279В,	f=2750.0000Гц,	Pa=0.001631955054Вт
t=6.00с:	u=51.4920В,	f=2750.0000Гц,	Pa=0.001466029649Вт
t=6.50с:	u=48.5030В,	f=2750.0000Гц,	Pa=0.001300765189Вт
t=7.00с:	u=45.4519В,	f=2750.0000Гц,	Pa=0.001142264569Вт
t=7.50с:	u=42.4076В,	f=2750.0000Гц,	Pa=0.000994373633Вт
t=8.00с:	u=39.4208В,	f=2551.2946Гц,	Pa=0.000797150836Вт
t=8.50с:	u=36.5281В,	f=2366.9469Гц,	Pa=0.000634996922Вт
t=9.00с:	u=33.7550В,	f=2195.9196Гц,	Pa=0.000503062696Вт
t=9.50с:	u=31.1184В,	f=2037.2501Гц,	Pa=0.000396650222Вт
t=10.00с:	u=28.6283В,	f=1890.0455Гц,	Pa=0.000311453804Вт

Розрахунки завершено, збережено у файлі 'results.txt'

Рисунок 9 – Результат роботи програми (3 частина)

Графіки

Для візуалізації результатів роботи програми було побудовано графік (Рисунок 10) залежності активної потужності P_a від часу t для кожного варіанту розрахунків. Враховуючи експоненційний характер зміни параметрів та малі значення потужності, для побудови графіків було обрано напівлогарифмічний масштаб, що дозволило наочно відобразити динаміку процесу.

Графік $P_a(t)$ (Значення Y наближені)]

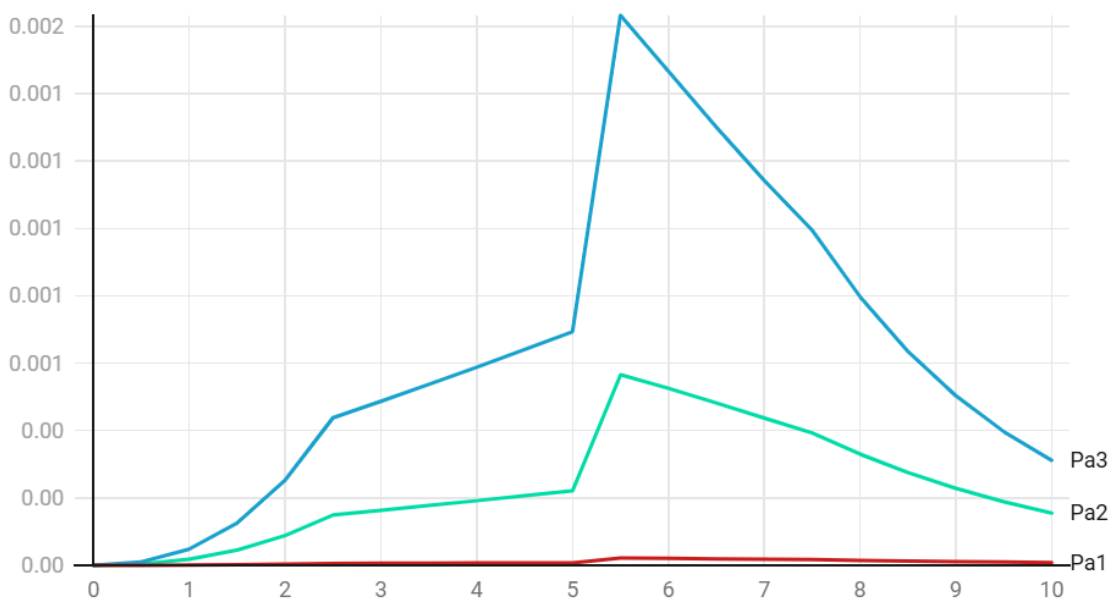


Рисунок 10 – Графік результатів програми

Висновки

Під час виконання курсової роботи було реалізовано комплексне дослідження процесів розсіювання активної потужності в діелектрику конденсатора при змінних параметрах електричного поля. Основна цінність роботи полягає у створенні програми на мові C, яка поєднує теоретичні розрахунки з практичною реалізацією. Під час розробки було успішно застосовано різні можливості мови програмування C, зокрема роботу зі структурами даних для ефективної організації вхідних параметрів та функціональний підхід з чітким розподілом обчислювальних задач між окремими модулями програми.

Важливим аспектом реалізації стала робота з файловим введенням/виведенням, що дозволило організувати зручний спосіб зчитування вхідних даних та збереження результатів обчислень. Для виконання складних математичних розрахунків було використано стандартні математичні функції, що забезпечило високу точність обчислень навіть при роботі з дуже малими величинами. Створення власних функцій значно полегшило взаємодію з кодом як для розробника, так і для читача.

Проведена робота наочно демонструє потенціал мови C для вирішення складних науково-технічних завдань і може служити основою для подальших досліджень у цій галузі.

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.