

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

**СИСТЕМИ ТА МЕТОДИ ВІЗУАЛІЗАЦІЇ ФІЗИЧНИХ РОЗРАХУНКІВ  
НА ПРИКЛАДІ ДИСПЕРСІЇ ЗВОРОТНИХ ОБ'ЄМНИХ  
МАГНІТОСТАТИЧНИХ ХВИЛЬ**

Дипломна робота бакалавра

студента 4 року навчання

Спеціальність: 123 «Комп'ютерна інженерія»

ОНП «Комп'ютерна системи та мережі»

**Родіона БАКАЄВА**

Науковий керівник:

канд. фіз.-мат. наук

**Олексій НЕЧИПОРУК,**

доцент кафедри квантової радіофізики

та наноелектроніки

Рецензент

канд. фіз.-мат. наук В'ячеслав БОРЕЦЬКИЙ,

доцент кафедри радіотехніки та радіоелектронних систем

До захисту допускаю:

Завідувач кафедрою

Юрій БОЙКО /

Ухвалено на засіданні кафедри “\_\_\_\_\_” \_\_\_\_\_ 2022 р., протокол № \_\_\_\_\_

**Київ - 2022**

## РЕФЕРАТ

Звіт з науково-виробничої практики з комп'ютерної інженерії: 35 с., 10 рис., 4 джерела, 5 додатків.

Аналізується дисперсія зворотних об'ємних магнітостатичних хвиль (ЗОМСХ) в ізолюваному феритовому шарі. Отримано та проаналізовано відповідне дисперсійне співвідношення. Написано програми для візуалізації дисперсії ЗОМСХ.

**КЛЮЧОВІ СЛОВА:** хвилі магнітостатичні, дисперсія, МСХ, ЗОМСХ, Wolfram, MATLAB, Scilab, WPF.

## Зміст

Вступ.....	3
1. Дисперсійні залежності в МСХ .....	5
1.1 Магнітостатичні хвилі в ізолюваному феритовому шарі.....	7
1.2 Зворотні об'ємні МСХ.....	8
2. Системи візуалізації фізичних розрахунків.....	15
2.1 Wolfram Mathematica.....	15
2.2 MATLAB та Scilab.....	17
3. Комп'ютерне моделювання дисперсії ЗОМСХ.....	19
Висновки.....	21
Перелік посилань.....	22
Додатки.....	23

## ВСТУП

На даному етапі науково-технічного прогресу вимагається створення великої кількості складної радіоелектронної апаратури, основними вимогами для якої є надійність, економічність на відносно невелика собівартість. Через збільшення ступеня інтеграції мікросхем відбувається ріст функціональної та конструктивної складності апаратури при збереженні принципу створення електросхем зі стандартних радіоелементів (транзисторів, конденсаторів, резисторів, діодів і т.д.). Але через збільшення інтеграції і з цього зменшення розмірів мікросхем та їх складових має певні фізичні обмеження.

Для розв'язання задачі обмеження фізичних розмірів функціональна електроніка пропонує новий підхід розробки мікросхем, що дає змогу реалізувати обробку електромагнітних сигналів не застосовуючи базові радіоелектронні елементи. Принципи даного підходу базуються на безпосередньо фізичних явищах, що протікають у твердому тілі. В розглянутому випадку саме певні властивості твердого тіла можуть допомогти при виконанні певної функції.

Через це дослідники звернули увагу на магнітостатичні хвилі (МСХ) в феритових монокристалах. Справа в тому, що серед типів елементарних збуджень твердих тіл МСХ мають унікальне розмаїття лінійних і нелінійних властивостей, а саме, дисперсійних характеристик за допомогою яких можна керувати різними процесами. Ще однією з переваг МСХ є малі втрати при перетворенні і поширенні в НВЧ діапазоні.

Всі ці фактори є стимулом для формування нового напрямку функціональної електроніки, а саме - спін-хвильової електроніки НВЧ. На відміну від схемотехнічних та цифрових рішень даний напрямок дає змогу конструювати планарні пристрої та системи обробки електричних сигналів у масштабі реального часу безпосередньо на несучій частоті.

Властивості МСХ дають змогу створювати різноманітні прилади НВЧ-електроніки з великою варіацією параметрів. Затримка МСХ керується шляхом використання багатой кількості шарів ферит діелектричних-структур

та шляхом формування неоднорідності зовнішнього магнітного поля, що було експериментально та фізично доведено. З цього боку дослідження МСХ у неоднорідних феритових плівках з неперервною зміною матеріальних параметрів згідно довжини плівки є доволі перспективним.

На даний момент рівень розвитку функціональної електроніки потребує точності у вимірюванні характеристик МСХ в феритових структурах з реальними параметрами. Таким чином в даній роботі поставлена мета дослідити дисперсію ЗОМСХ (зворотних об'ємних магнітостатичних хвиль) в структурі з двома нарами фериту.

# 1. Дисперсійні залежності МСХ

## 1.1 Магнітостатичні хвилі в ізолюваному феритовому шарі

Розглянемо властивості МСХ в ізолюваному феритовому шарі. Таку структуру було обрано через простоту, також так модель являється гарним наближенням для плівкових монокристалічних матеріалів.

В рамках магнітостатичного наближення, розв'язуючи рівняння Уокера розглянемо МСХ, які поширюються під довільним кутом  $\theta_k$  до  $\vec{M}_0$  (рис.1.3) в дотично намагніченому феритовому шарі.

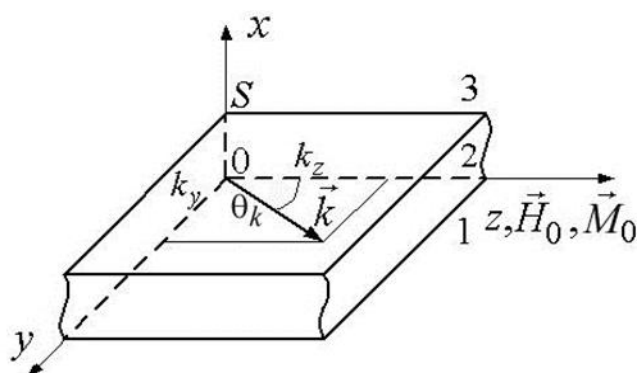


Рис.1.3 Дотично-намагнічений необмежений феритовий шар.

Магнітні властивості діелектричних шарів описуємо тензором:

$$\mu_{ij} = \delta_{ij}, \quad (\delta_{ij} - \text{символ Кронеккера}).$$

Система рівнянь, яка описує магнітостатичний потенціал у нашій структурі:

$$\Psi = \begin{cases} \Psi_1, & x < 0, \\ \Psi_2, & 0 < x < s, \\ \Psi_3, & x > s. \end{cases} \quad (1.1)$$

З системи рівнянь 1.1 задача зводиться до розв'язку рівнянь:

$$\Delta\Psi_1 = 0, \quad x < 0, \quad (1.2)$$

$$\mu \left( \frac{\partial^2 \Psi_2}{\partial x^2} + \frac{\partial^2 \Psi_2}{\partial y^2} \right) + \frac{\partial^2 \Psi_2}{\partial z^2} = 0, \quad 0 < x < s, \quad (1.3)$$

$$\Delta\Psi_3 = 0, \quad x > s. \quad (1.4)$$

Рівняння (1.2) та (1.4) еліптичного типу, а рівняння (1.3) в залежності від знака параметра  $\mu$  може бути як еліптичного ( $\mu > 0$ ), так і гіперболічного типу ( $\mu < 0$ ). В залежності від умови  $\mu > 0$  розв'язки рівняння (2.3) являють собою поверхневі, а при  $\mu < 0$  – зворотні об'ємні хвилі.

Розв'язки рівнянь (1.2)–(1.4) у вигляді неоднорідних плоских хвиль, хвильовий вектор яких  $\vec{k}_s = (0, k_y, k_z)$  знаходиться в площині ZY:

$$\Psi_i = X_i(x) e^{i(\omega t - k_y y - k_z z)}, \quad i = 1, 3. \quad (1.5)$$

Підстановка виразів (1.5) в рівняння (1.2)–(1.4) призводить до наступних звичайних диференціальних рівнянь з постійними коефіцієнтами:

$$\frac{d^2 X_i}{d x^2} - k_s^2 X_i = 0, \quad i = 1, 3, \quad (1.6)$$

$$\frac{d^2 X_2}{d x^2} - \kappa^2 X_2 = 0, \quad (1.7)$$

де:

$$k_s = \sqrt{k_y^2 + k_z^2}, \quad \kappa = \sqrt{\frac{k_z^2}{\mu} + k_y^2}, \quad k_y = k \sin \theta_k, \quad k_z = k \cos \theta_k. \quad (1.8)$$

## 1.2 Зворотні об'ємні МСХ

В цьому випадку напрям поширення хвилі співпадає з напрямом зовнішнього магнітного поля.

Нехай стала поширення  $\kappa$  у виразі (1.8) є дійсною додатною величиною. Так твердження виконується за умови, що  $\mu > 0$  або (оскільки

$$\mu = \frac{\omega^2 - \omega_1^2}{\omega^2 - \omega_H^2} \text{ та } \omega_1 > \omega_H), \text{ якщо } \omega < \omega_H \text{ і } \omega > \omega_1.$$

Враховуючи обмеженість магнітостатичних потенціалів  $\psi_1$  і  $\psi_3$  на нескінченості маємо вирази для магнітостатичних потенціалів

$$\Psi_1 = Ae^{k_s x + i(\omega t - k_y y - k_z z)} \quad (1.9)$$

$$\Psi_2 = (B \operatorname{ch} \kappa x + C \operatorname{sh} \kappa x) e^{i(\omega t - k_y y - k_z z)} \quad (1.10)$$

$$\Psi_3 = De^{-k_s x + i(\omega t - k_y y - k_z z)} \quad (1.11)$$

Використаємо граничні умови тангенційних компонент напруженості магнітного поля і нормальних компонент магнітної індукції на межах середовищ для знаходження дисперсійного співвідношення для МХС:

$$\begin{aligned} \Psi_1|_{x=0} &= \Psi_2|_{x=0}, \quad \frac{\partial \Psi_1}{\partial x}|_{x=0} = \mu \frac{\partial \Psi_2}{\partial x}|_{x=0} - i\mu_a \frac{\partial \Psi_2}{\partial y}|_{x=0}, \\ \Psi_2|_{x=s} &= \Psi_3|_{x=s}, \quad \mu \frac{\partial \Psi_2}{\partial x}|_{x=s} - i\mu_a \frac{\partial \Psi_2}{\partial y}|_{x=s} = \frac{\partial \Psi_3}{\partial x}|_{x=s}, \end{aligned}$$

Підставивши вирази для магнітостатичних потенціалів (1.9)-(1.11)

Отримуємо систему рівнянь для констант А,В,С,Д:

$$\begin{cases} A = B \\ De^{-k_s S} = B \operatorname{ch} \kappa S + C \operatorname{sh} \kappa S \\ Ak_s = C\mu\kappa - B\mu_a k_y \\ -Dk_s e^{-k_s S} = -B\mu\kappa \operatorname{sh} \kappa S + C\mu\kappa \operatorname{ch} \kappa S - \\ -B\mu_a k_y \operatorname{ch} \kappa S - C\mu_a k_y \operatorname{sh} \kappa S \end{cases} \quad (1.12)$$

Розкривши рівняння власних хвиль, визначник яких повинен дорівнювати нулю, виводимо дисперсійне співвідношення в неявній формі:

$$2\mu\kappa k_s \operatorname{cth} \kappa S + k_s^2 - \mu_a^2 k_y^2 - \mu^2 \kappa^2 = 0 \quad (1.13)$$

Підставивши замість сталої поширення  $\kappa$  вираз (1.8) маємо:

$$2k_s \sqrt{\mu(k_z^2 + \mu k_y^2)} \operatorname{cth} \sqrt{k_y^2 + \frac{k_z^2}{\mu}} s + \mu(k_z^2 + \mu k_y^2) + k_s^2 - \mu_a^2 k_y^2 = 0, \quad (1.14)$$

З рівняння (1.14) будемо спектр МСХ у дотично-намагніченому феритовому шарі:

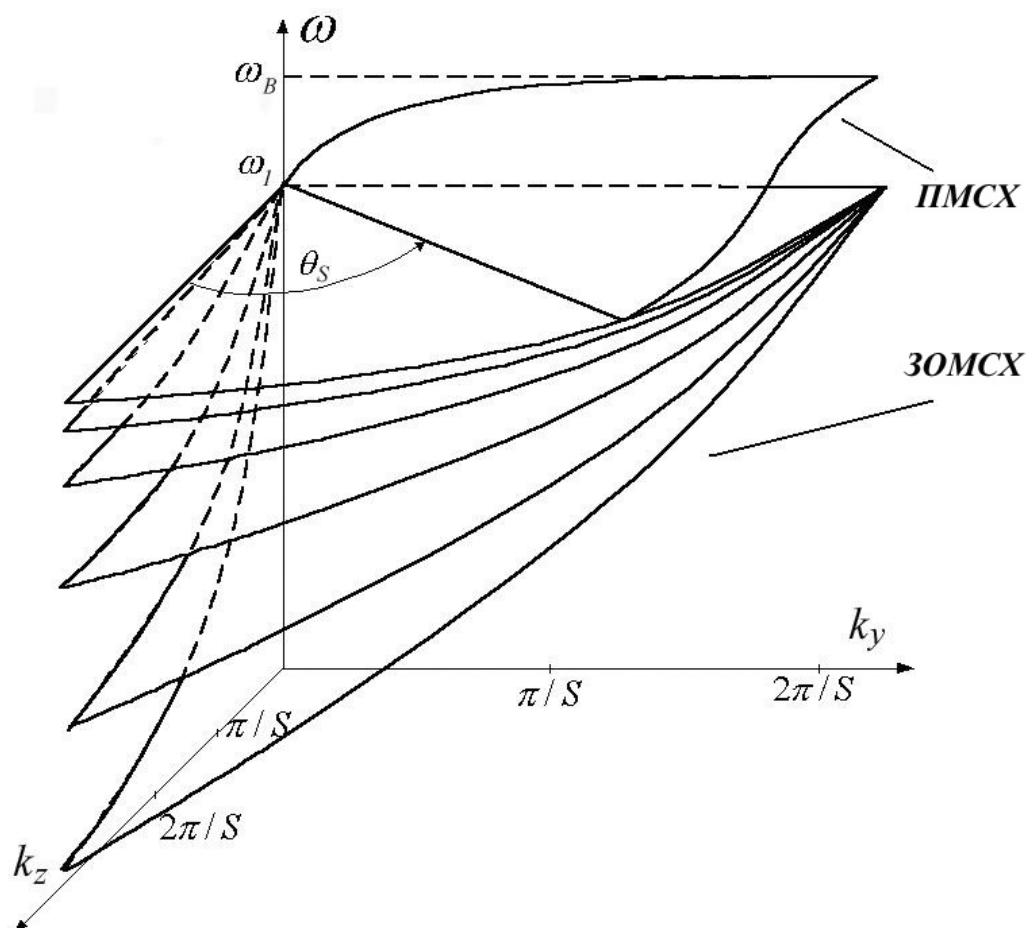


Рис.1.4 Спектр магнітостатичних хвиль у дотично-намагніченому феритовому шарі.

З пп 1.2 зворотні об'ємні МСХ (ЗОМСХ) є розв'язками рівняння(1.7), що відповідають уявним величинам площини  $\kappa$ . Таке твердження справджується за умови  $\mu < 0$ , тобто, в діапазоні частот  $\omega_H < \omega < \omega_1$ .

Дисперсійне співвідношення для ЗОМСХ можемо дістати (1.13):

$$\kappa_0 = \sqrt{-\left(\frac{k_z^2}{\mu} + k_y^2\right)} \quad (1.9)$$

Дане співвідношення є дійсною величиною, то му враховуючи, що  $cth i\kappa_0 s = -i ctg \kappa_0 s$  маємо дисперсію зворотних об'ємних магнітостатичних хвиль:

$$2k_s \mu \kappa_0 ctg \kappa_0 s - \mu^2 \kappa_0^2 + k_s^2 - \mu_a^2 k_y^2 = 0 \quad (1.10)$$

Спектр ОМСХ (рис. 1.4) лежить нижче за спектр поверхневих МХС і займає область частот, що визначається виразом:

$$\omega_H < \omega < \omega_1 \quad (1.11)$$

Набір відповідних поверхонь, згідно рис 1.4 відповідає кореням (1.10). Із співвідношення (1.10) випливає, що дисперсія ЗОМСХ не змінюється при зміні напрямку поширення хвиль на протилежний.

Для подальшого аналізу розглянемо ЗОМСХ. Ці хвилі поширюються вздовж вектора намагніченості  $\vec{M}_0$  (тобто,  $\theta_k = 0$ ). Поклавши у вираз (1.9)  $k_y = 0$  (впливаючи з цього,  $\partial\Psi/\partial y = 0$ ), знаходимо:

$$2ctg \frac{|k_z|s}{\sqrt{-\mu}} = \frac{1}{\sqrt{-\mu}} - \sqrt{-\mu} \quad (1.12)$$

Зауважимо, що відповідна система рівнянь для опису електродинамічних граничних умов у випадку ЗОМСХ має вигляд:

$$\Psi_1|_{x=0} = \Psi_2|_{x=0}, \quad \Psi_2|_{x=s} = \Psi_3|_{x=s},$$

$$\frac{\partial\Psi_1}{\partial x}|_{x=0} = \mu \frac{\partial\Psi_2}{\partial x}|_{x=0}, \quad \mu \frac{\partial\Psi_2}{\partial x}|_{x=s} = \frac{\partial\Psi_3}{\partial x}|_{x=s}.$$

Використовуючи формулу котангенса подвійного аргументу, дисперсійне співвідношення (2.28) можемо переписати так:

$$\left( \operatorname{tg} \frac{|k_z|s}{2\sqrt{-\mu}} - \sqrt{-\mu} \right) \left( \operatorname{tg} \frac{|k_z|s}{2\sqrt{-\mu}} + \frac{1}{\sqrt{-\mu}} \right) = 0,$$

звідки

$$|k_{zn}|s = 2\sqrt{-\mu} \left( n\pi + \operatorname{arctg} \sqrt{-\mu} \right), \quad (1.13)$$

де  $n = 0, 1, 2, \dots$  та

$$|k_{zm}|s = 2\sqrt{-\mu} \left( m\pi - \operatorname{arctg} \frac{1}{\sqrt{-\mu}} \right), \quad (1.14)$$

де  $m = 1, 2, 3, \dots$

З цього можна зробити висновок, що спектр ЗОМСХ складається з нескінченного числа гілок, що відповідають різним значенням  $m$  та  $n$ , а саме симетричним та антисиметричним модам відповідно. Кожна з гілок займає певну область частот (1.11), при цьому  $k = 0$  на частотах  $\omega = \omega_1$ , а  $k \rightarrow \infty$  при  $\omega \rightarrow \omega_H$  (рис.2.7).

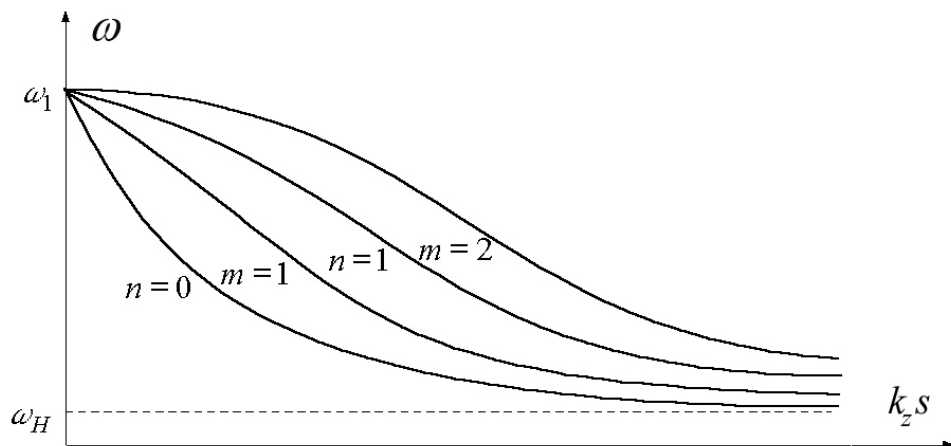


Рис. 1.5. Дисперсія зворотних об'ємних МСХ, які поширюються вздовж напрямку вектора намагніченості  $\vec{M}_0$  ( $\theta_k = 0$ )

Частотний діапазон, який займають ЗОМСХ, дорівнюють:

$$\Delta\omega_V = \omega_1 - \omega_H = \frac{\omega_M}{\sqrt{1 + \frac{\omega_M}{\omega_H} + 1}}$$

Звідси робимо висновок, що при  $\omega_H \rightarrow +\infty$  маємо  $\Delta\omega_V \rightarrow \omega_M/2$ . Діапазон існування ЗОМСХ при збільшенні зовнішнього магнітного поля не звужається до нуля, а прямує до постійного значення, яке при  $\omega_H = \infty$  складає  $\Delta\omega = \omega_M/2$ .

Проаналізуємо основну, або фундаментальну, моду ЗОМСХ, фкі відповідає, що є найнижчою по частоті. Фундаментальна мода має найпростіший розподіл змінної намагніченої за координатою  $x$  і через це легко збуджується. Слід зауважити, що найнижчою по частоті модою ЗОМСХ є антисиметрична мода. На початковій ділянці дисперсійної кривої групова швидкість даної моди рівна величині:

$$V_{gp} \cong -\frac{s}{4} \frac{\omega_H \omega_M}{\omega_1} \operatorname{sgn} k_z$$

Маємо справу саме із зворотними хвилями, про це свідчить знак мінус. Для зворотних хвиль вектори групової та фазової швидкостей протилежні, через це їх і називають зворотними.

Розглянемо розподіл магнітного поля ЗОМСХ. Поклавши в системах рівнянь (1.9)-(1.11)  $k_y = 0$ ,  $k_z \equiv k$ ,  $\kappa = ik / \sqrt{-\mu}$  отримуємо вирази для магнітостатичних потенціалів:

$$\Psi_1 = A e^{kx+i(\omega t-kz)}, \quad (1.15)$$

$$\Psi_2 = A \left( \cos \frac{kx}{\sqrt{-\mu}} - \frac{1}{\sqrt{-\mu}} \sin \frac{kx}{\sqrt{-\mu}} \right) e^{i(\omega t-kz)}, \quad (1.16)$$

$$\Psi_3 = A \left( \cos \frac{kx}{\sqrt{-\mu}} - \frac{1}{\sqrt{-\mu}} \sin \frac{ks}{\sqrt{-\mu}} \right) e^{k(s-x)+i(\omega t-kz)}. \quad (1.17)$$

Далі розглянемо окремо хвилі, що описуються дисперсійними співвідношеннями (1.13) (1.14). Ці два типи дисперсії відповідають симетричним і антисиметричним по змінній  $x$  розв'язкам рівнянь (1.15)-(1.17) відносно середини феритового шару, тобто точки  $x = s/2$ .

**Антисиметричні моди.** Використовуючи дисперсійне співвідношення для антисиметричних мод (1.13), приводимо систему рівнянь (1.15)–(1.17) до наступного вигляду:

$$\begin{aligned}\Psi_1 &= Ae^{kx+i(\omega t-kz)}, \\ \Psi_2 &= -\frac{A}{\sin \frac{ks}{2\sqrt{-\mu}}} \sin \frac{k}{\sqrt{-\mu}} \left(x - \frac{s}{2}\right) e^{i(\omega t-kz)}, \\ \Psi_3 &= -Ae^{k(s-x)+i(\omega t-kz)}.\end{aligned}$$

З даного співвідношення виходить, що  $\Psi_i$  – антисиметрична функція за координатою  $x$ . Тепер ми маємо змогу визначити компоненти поля у відповідних областях:

$$\tilde{h}_x = k\Psi_1, \quad \tilde{h}_z = -ik\Psi_1, \quad x < 0, \quad (1.18)$$

$$\tilde{h}_x = -\frac{Ak}{\sqrt{-\mu} \sin \frac{ks}{2\sqrt{-\mu}}} \cos \frac{k}{\sqrt{-\mu}} \left(x - \frac{s}{2}\right) e^{i(\omega t-kz)}, \quad \tilde{h}_z = -ik\Psi_2, \quad 0 < x < s, \quad (1.19)$$

Використовуючи дані рівняння визначимо наступні вирази для силових ліній магнітного поля:

$$x = -\frac{1}{k} \ln |\sin(\omega t - kz)| + C, \quad x < 0, \quad (1.20)$$

$$\left| \cos \frac{k}{\sqrt{-\mu}} \left(x - \frac{s}{2}\right) \right|^{-\mu} = |C \sin(\omega t - kz)|, \quad 0 < x < s, \quad (1.21)$$

$$x = \frac{1}{k} \ln |\sin(\omega t - kz)| + C, \quad x > s. \quad (1.22)$$

Виходячи з рівнянь (1.20)-(1.22) ми можемо побудувати схему траєкторій ліній магнітного поля для моди  $n=0$  при  $t=0$  і  $ks = \pi/2$ .

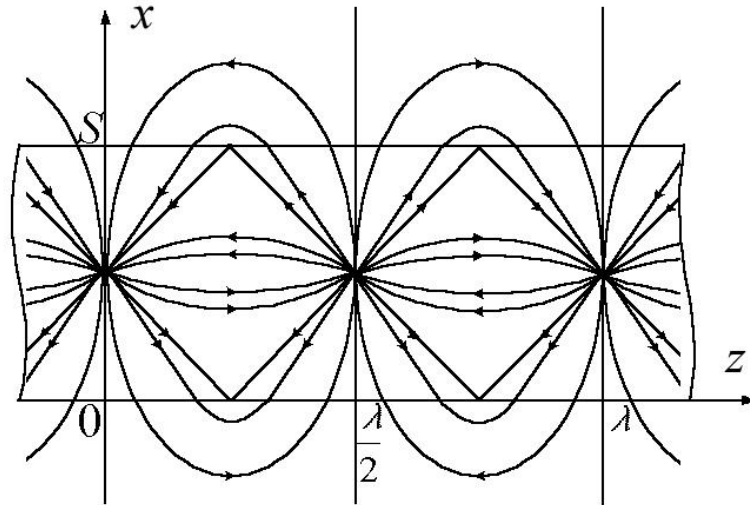


Рис.1.6. Розподіл силових ліній магнітного поля зворотних об'ємних МСХ для моди  $n=0$

Використовуючи відповідні вирази для компонент напруженості змінних магнітних полів ЗОМСХ, можемо навести їх розподіл за координатою  $x$ .

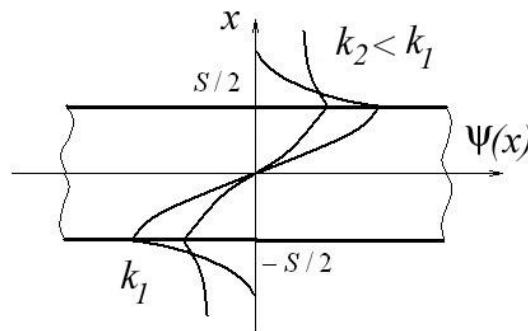


Рис. 1.7. Розподіл амплітуд НВЧ компонент магнітного поля ЗОМСХ моди для різних значень хвильового числа

**Симетричні моди.** Використовуючи дисперсійне співвідношення для симетричних мод (1.19), можна привести систему рівнянь (1.20)–(1.22) до наступного вигляду:

$$\Psi_1 = Ae^{kx+i(\omega t-kz)} , \quad (1.23)$$

$$\Psi_2 = \frac{A}{\cos \frac{ks}{2\sqrt{-\mu}}} \cos \frac{k}{\sqrt{-\mu}} \left( x - \frac{s}{2} \right) e^{i(\omega t-kz)} , \quad (1.24)$$

$$\Psi_3 = Ae^{k(s-x)+i(\omega t-kz)} . \quad (1.25)$$

Тут хвильове число визначається з дисперсії (2.30), а  $\Psi_i$  є симетричною функцією за координатою  $x$ .

## 2. Системи візуалізації фізичних розрахунків

### 2.1 Wolfram Mathematica

Wolfram Mathematica – це програмне забезпечення (ПЗ) в основі якої лежить система комп'ютерної алгебри розроблена компанією Wolfram Research. Wolfram Mathematica є Desktop ПЗ, що підтримує роботу на Microsoft Windows, macOS, Linux.

Дане ПЗ має досить широкий функціонал для аналітичних перетворень, чисельних розрахунків, роботи з графікою, включно з можливістю побудови двовимірних та тривимірних графіків функцій.

Можливості з чисельних розрахунків:

- Обчислення простих арифметичних функцій
- Обчислення значень функції
- Знаходження розв'язку системи рівнянь
- Поліноміальна інтерполяція

Можливості з лінійної алгебри:

- Всі можливі операції з матрицями
- Пошук власних значень та значень векторів

Можливості з аналітичних перетворень:

- Розв'язання систем рівнянь та нерівностей, у тому числі поліноміальних та тригонометричних
- Спрощення рівнянь та виразів
- Знаходження границь змінних
- Знаходження значень інтегрованих та диференційованих функцій

Можливості взаємодії з графікою:

- Побудова геометричних фігур
- Побудова графіків функцій

- Побудова графів та взаємодія з ними

Також Wolfram Mathematica підтримує функціональну мову програмування Wolfram, розробленою компанією Wolfram Research спеціально для системи Wolfram Mathematica. Wolfram має достатньо великий функціонал з різних сфер, у тому числі і для моделювання ЗОМСХ.

Також існує web-аналог Wolfram Mathematica, але з меншою кількістю функціоналу, - WolframAlpha.

Приклад коду та виведення графіку співвідношення для наданий у Додатку 1 і 2 відповідно.

## 2.2 MATLAB та Scilab

MATLAB – система, що складається з прикладних програм для числового аналізу та високорівневої мови програмування. Даний пакет програм створений компанією The MathWorks. Пакет є Desktop-додатком для Windows, Linux та MAC OS X.

MATLAB має зручний функціонал для роботи з математичними матрицями, функціями, алгоритмами, моделюванням графіків та створенням робочих інтерфейсів з програмами написаних іншими мовами програмування.

Функціонал MATLAB пориває майже всі області математики, зокрема:

- Лінійна алгебра та алгебра матриць
- Розрахунок многочленів та інтерполяція
- Статистичний аналіз та обробку даних
- Диференційні та інтегральні рівняння
- Цілочисельна арифметика
- Побудова графіків

Scilab – пакет програм призначений для чисельних обчислень, розроблений вченими INRIA для. Пакет є Desktop-додатком для Windows, Linux та MAC OS X.

Scilab, як і MATLAB надає середовище для інженерних та наукових розрахунків. Також ці 2 пакети мають схожу мову програмування та функціонал для роботи.

Функціонал Scilab:

- Лінійна алгебра
- Побудова 2d- та 3d-графіки
- Інтерполяція многочленів
- Статистичний аналіз

### Порівняння MATLAB та Scilab:

MATLAB	Scilab
Мова програмування високого рівня, яка використовується для виконання математичних обчислень.	Це програмне забезпечення, яке використовується для виконання наукових обчислень.
Ця мова написана на C, C++ і Java.	Це програмне забезпечення запрограмовано на C, C++ і Fortran.
Збережений файл має розширення «.m».	Збережений файл має розширення «.sci».
Командний рядок починається з «%».	Командний рядок починається з «//».
Це не open-source ПЗ.	Це open-source ПЗ.
MATLAB використовується для вирішення високорівневих обчислень.	Scilab використовується для вирішення низькорівневих наукових обчислень.

Для прикладу роботи розглянемо програму Scilab, оскільки це програмне забезпечення є open-source. У Scilab ми розглянемо побудови геометричної фігури, а саме сфери. Для цього ми задамо дві матриці з значеннями та формули для координат  $x, y, z$ .

Лістинг коду програми та результат її виконання надано у Додатках 3 і 4 відповідно.

### 3. Комп'ютерне моделювання дисперсії ЗОМСХ

Метою роботи є розробка програми, що дозволяє змоделювати дисперсію ЗОМСХ в тривимірному просторі, координатами якого слугують хвильові числа та частоти. Головною метою програми є візуалізація дисперсії, що в свою чергу забезпечує можливість проведення візуального аналізу. Також візуалізація дисперсійних залежностей у тривимірному просторі з методичної точки зору сприяє наглядному розумінню фізики МСХ та принципів роботи приладів спін-хвильової електроніки студентами.

З цією метою було розроблено WPF застосунок з використанням бібліотек Media3D та HelixToolKit. Інтерфейс програми містить наступні поля: область вводу та виводу цифрових даних та область відображення графічної 3D моделі.

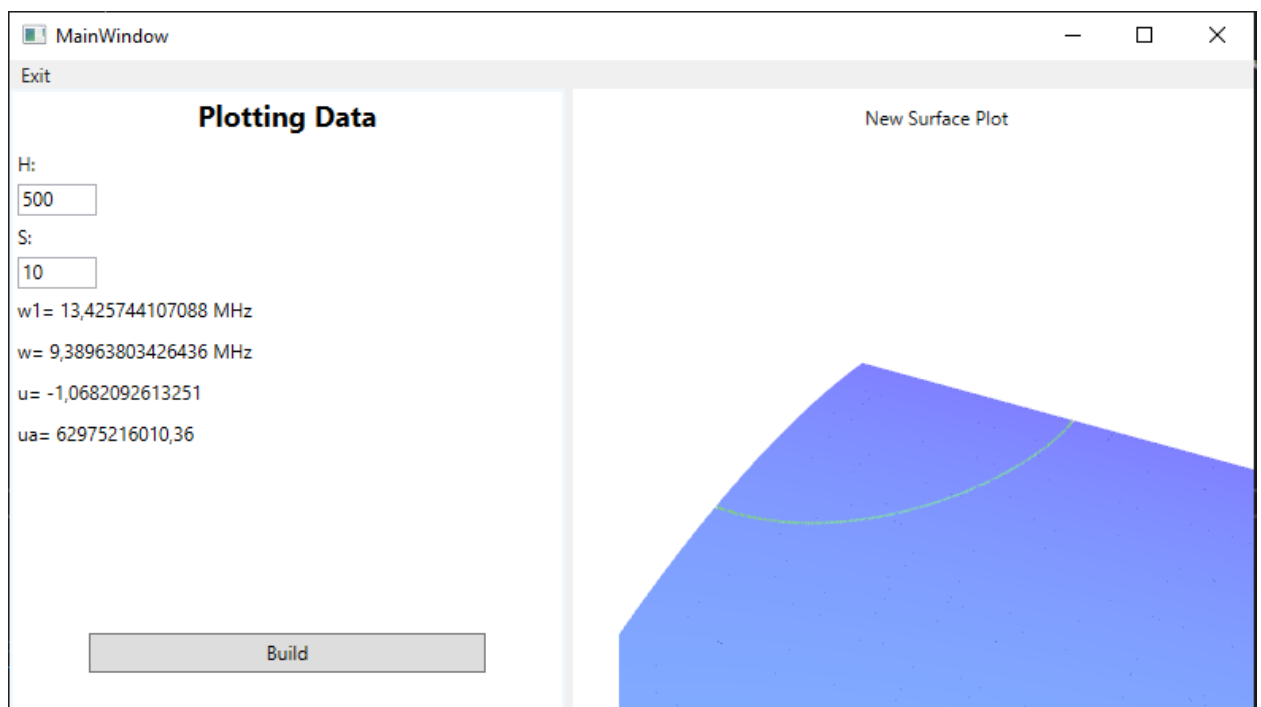


Рис.3.1 Інтерфейс програми для обрахунку дисперсії ЗОМСХ

Область вводу/виводу цифрових даних. Цю область розташовано вертикально у лівій частині вікна програми. В ній знаходяться наступні поля вводу: «Magnetizing Field» – задається величина зовнішнього магнітного поля та товщина прошарків діелектрика; кнопка «Build», що обчислює значення дисперсійного рівняння та відображує ці значення у вигляді тривимірного зображення у відповідній області.

Область відображення тривимірного представлення. В цій області знаходиться площина ( ); значення та , що обмежені площиною, підбираються в залежності від товщини феритового шару. Перпендикулярно до центру площини проведено вісь для відображення частот ( $\omega$ ).

Керування інтерфейсом здійснюється за допомогою миші та клавіатури. З їх допомогою можна вводити дані, активізувати процес обчислення, а також забезпечувати обертання зображень вздовж вертикальної і горизонтальної вісей екрану.

Мінімальні вимоги до обчислювальної техніки, на якій можливе встановлення та задовільна робота програми наступні: Windows 7, процесор з частотою 1,8 ГГц, 1 Гб оперативної пам'яті, наявність бібліотек HelixToolKit та Visual Studio.

## ***Висновки***

1. Розглянуто основні фізично-хімічні властивості ЗІГ, як матеріалу поширення МСХ
2. Розглянуто основні властивості та залежності поширення МСХ в ізольованому феритовому шарі, як моделі з найпростішою структурою.
3. Отримано дисперсійне співвідношення для зворотних об'ємних магнітостатичних хвиль в структурі з двома шарами фериту. За допомогою граничних переходів було здійснено його перевірку за випадком ізольованого феритового шару.
4. Розглянуто деякі готові програмні рішення для математичних, фізичних розрахунків та їх візуалізації.
5. Створено програму, що графічно відображає дисперсійні залежності ЗОМСХ.

### *Перелік посилань*

- [1] Гуревич А. Г., Мелков Г. А. Магнитостатические колебания и волны. –М.: Физматлит, 1994. – 464 с.
- [2] Гуляев Ю. В., Зильберман П.Е. Спинволновая электроника. – М.: Знание, 1988. – 54 с.
- [3] Данилов В.В., Зависляк І.В., Нечипорук О.Ю. Спін-хвильова електродинаміка. - Киев: ВПЦ «Київський університет», 2008. - 351 с.
- [4] В.Д. Федорів, Н.В. Сташко, І.П. Яремій, В.В. Мокляк Формування структури залізо-ітрієвого гранату на основі вихідного матеріалу, отриманого золь-гель методом
- [5] Wolfram T. Magnetostatic surface waves in layered magnetic structures // J. Appl. Phys. – 1970. – V. 41. – P. 4748-4749.
- [6] Pfeifer H. Characteristic of magnetostatic surface waves for a system of two magnetic films // Phys. Stat. Sol. (a). – 1973. – V. 18 – №1 – P. K53-K56.

## Додатки

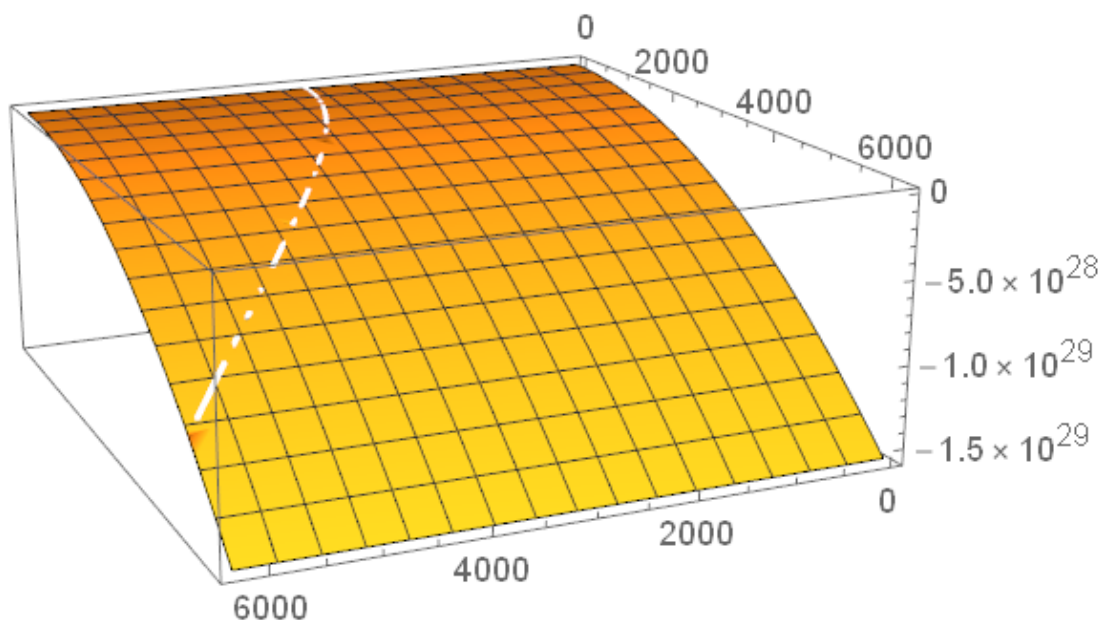
### Додаток 1 (Лістинг коду *Walfram Mathematica*)

```

γ = 2800000;
M = 1750;
H0=1000;
S = 0.001;
wh = γ*H0;
wm=γ*M*4*Pi;
w1 = wh*(wh+wm);
w=γ*2*Pi*M;
u= (w^2-w1)/(w^2-wh^2);
ua=(w*wm)/(w^2-wh^2);
k0=Sqrt[-(kz^2/u +ky^2)];
ks= Sqrt[ky^2+kz^2];
Plot3D[2*ks*u*k0* Cot[k0*S]- u^2*k0^2+ks^2-
ua^2*ky^2,{kz,0,(2*Pi)/S},{ky,0,(2*Pi)/S}]

```

### Додаток 2 (Результат виконання коду у *Walfram Mathematica*)



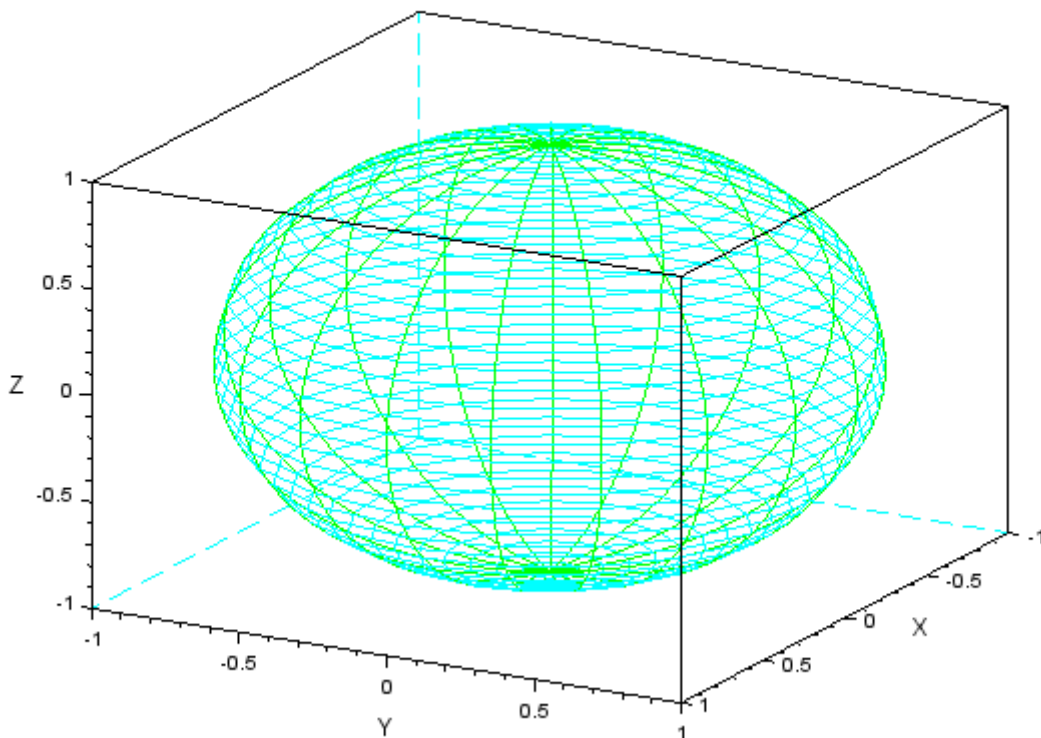
### Додаток 3 (Лістинг коду Scilab)

```

u = linspace(-%pi/2, %pi/2, 40);
v = linspace(0, 2*%pi, 20);
x = cos(u)*cos(v);
y = cos(u)*sin(v);
z = sin(u)*ones(v);
plot3d3(x, y, z);

```

### Додаток 4 (результат роботи програми у Scilab)



### Додаток 5 (лістинг коду програми)

Main.xaml:

```

<Window x:Class="WpfDiplom2.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

```

```

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:SurfacePlot="clr-namespace:WPFSurfacePlot3D"
xmlns:local="clr-namespace:WpfDiplom2"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
<DockPanel x:Name="Parent">

    <Menu DockPanel.Dock="Top">

        <MenuItem Header="Exit" Click="FileExit_Click" />

    </Menu>

    <Grid>

        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="1*" MinWidth="350" />
            <ColumnDefinition Width="5" />
            <ColumnDefinition Width="3*" MinWidth="200" />
        </Grid.ColumnDefinitions>

        <Border Grid.Column="0" Margin="0" BorderBrush="AliceBlue"
BorderThickness="2" />
        <Border Grid.Column="1" Margin="0" BorderBrush="AliceBlue"
BorderThickness="2" />

        <ScrollViewer Grid.Column="0" VerticalScrollBarVisibility="Auto">
            <StackPanel Margin="0">

                <TextBlock Text="Plotting Data" FontSize="18" FontWeight="Bold"
HorizontalAlignment="Center" Margin="5" />
                <Label>Magnetizing Field:</Label>
                <TextBox x:Name="textBox1" MaxLength="5" Height="20" Width="50"
Margin="5,0,0,0" HorizontalAlignment="Left"></TextBox>
                <Label>S:</Label>
                <TextBox x:Name="textBox2" MaxLength="5" Height="20" Width="50"
TextChanged="textBox2_TextChanged" HorizontalAlignment="Left" Margin="5,0,0,0"/>

                <Label x:Name="labelW1"></Label>
                <Label x:Name="labelW"></Label>
                <Label x:Name="labelU"></Label>
                <Label x:Name="labelUa"></Label>
            </StackPanel>
        </ScrollViewer>

        <GridSplitter Grid.Column="1" Width="5" HorizontalAlignment="Stretch" />

        <SurfacePlot:SurfacePlotView
            x:Name="surfacePlotView"
            Grid.Column="2"
            Title="Sample Surface Plot" Margin="29,0,0,0"
        />
        <Button x:Name="Button1" Width="250" Height="25" Content="Build"
Margin="50,300,50,25" Click="BuildFunction" VerticalAlignment="Bottom"/>
    </Grid>

</DockPanel>
</Window>

```

Main.xaml.cs:

```

using System;
using System.Windows;

```

```

using System.Windows.Controls;
using WPFSurfacePlot3D;

namespace WpfDiplom2
{
    public partial class MainWindow : Window
    {
        private SurfacePlotModel viewModel;

        public MainWindow()
        {
            InitializeComponent();
            viewModel = new SurfacePlotModel();
            surfacePlotView.DataContext = viewModel;
        }

        private void BuildFunction(object sender, RoutedEventArgs e)
        {
            long Y = 2800000;
            int M = 1750;
            long H0 = 1000;
            long H1 = Convert.ToInt32(textBox1.Text);
            double S = Convert.ToInt32(textBox2.Text) / 10000.0;
            double wm = Y * M * 4 * Math.PI;
            double wh0 = Y * H0;
            double wh1 = Y * H1;
            double w1 = Math.Sqrt(wh0 * (wh0 + wm));
            double w = Math.Sqrt(wh1 * (wh1 + wm));
            double u = (w * w - w1 * w1) / (w * w - wh1 * wh1);
            double ua = (w * w * wm) / (w * w - wh1 * wh1);
            labelW1.Content = "w1= " + (w1/Math.Pow(10,9)).ToString() + " MHz";
            labelW.Content = "w= " + (w / Math.Pow(10, 9)).ToString() + " MHz";
            labelU.Content = "u= " + u.ToString();
            labelUa.Content = "ua= " + ua.ToString();
            double[,] arrayOfPoints = new double[500, 500];
            double ky = 2 * Math.PI / S / 500;
            double kY = ky;
            for (int i = 0; i < 500; i++)
            {
                double kz = 2 * Math.PI / S / 500;
                double kZ = kz;
                for (int j = 0; j < 500; j++)
                {
                    double k0 = Math.Pow(-(kz * kz) / u + ky * ky, 1.0 / 2.0);
                    double ks = Math.Sqrt(ky * ky + kz * kz);

                    if (k0 != 0)
                        arrayOfPoints[i, j] = (2 * ks * u * k0 * (1 /
Math.Tan(k0 * S)) - u * u * (-(kz * kz) / (u + ky * ky)) - u * u * (-(kz * kz) / (u
+ ky * ky)) + ky * ky + kz * kz - ua * ua * ky * ky) / Math.Pow(10, 27);

                    if (Math.Round(k0 * S, 2) == Math.Round(Math.PI, 2))
                    {
                        arrayOfPoints[i, j] = -500;
                    }
                    kz += kZ;
                }
                ky += kY;
            }
        }
    }
}

```

```

        viewModel.PlotData(arrayOfPoints);
    }

    private void FileExit_Click(object sender, RoutedEventArgs e)
    {
        Close();
    }

    private void textBox2_TextChanged(object sender, TextChangedEventArgs e)
    {
    }
}
}

```

PlotModel.cs:

```

using HelixToolkit.Wpf;
using System;
using System.ComponentModel;
using System.Windows;
using System.Windows.Media;
using System.Windows.Media.Media3D;

namespace WPFSurfacePlot3D
{
    public enum ColorCoding
    {
        ByLights, ByGradientY
    }

    class SurfacePlotModel : INotifyPropertyChanged
    {
        private int defaultFunctionSampleSize = 100;

        public SurfacePlotModel()
        {
            Title = "New Surface Plot";
            XAxisLabel = "x-Axis";
            YAxisLabel = "y-Axis";
            ZAxisLabel = "z-Axis";

            ColorCoding = ColorCoding.ByLights;

            Func<double, double, double> sampleFunction = (x, y) => x-y;
            PlotFunction(sampleFunction, 0, 0.001);
        }

        public void PlotData(double[,] zData2DArray)
        {
            int n = zData2DArray.GetLength(0);
            int m = zData2DArray.GetLength(1);
            Point3D[,] newDataArray = new Point3D[n, m];
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {

```

```

        Point3D point = new Point3D(i, j, zData2DArray[i, j]);
        newDataArray[i, j] = point;
    }
}
dataPoints = newDataArray;
RaisePropertyChanged("DataPoints");
}

public void PlotData(Point3D[,] point3DArray)
{
}

public void PlotFunction(Func<double, double, double> function, double
minimumXY, double maximumXY)
{
    PlotFunction(function, minimumXY, maximumXY, minimumXY, maximumXY,
defaultFunctionSampleSize, defaultFunctionSampleSize);
}

public void PlotFunction(Func<double, double, double> function, double
xMinimum, double xMaximum, double yMinimum, double yMaximum, int xSampleSize, int
ySampleSize)
{
    xMin = xMinimum;
    xMax = xMaximum;
    yMin = yMinimum;
    yMax = yMaximum;

    double[] xArray = CreateLinearlySpacedArray(xMinimum, xMaximum,
xSampleSize);
    double[] yArray = CreateLinearlySpacedArray(yMinimum, yMaximum,
ySampleSize);

    DataPoints = CreateDataArrayFromFunction(function, xArray, yArray);
    switch (ColorCoding)
    {
        case ColorCoding.ByGradientY:
            ColorValues = FindGradientY(DataPoints);
            break;
        case ColorCoding.ByLights:
            ColorValues = null;
            break;
    }
    RaisePropertyChanged("DataPoints");
    RaisePropertyChanged("ColorValues");
    RaisePropertyChanged("SurfaceBrush");
}

private Point3D[,] CreateDataArrayFromFunction(Func<double, double, double>
f, double[] xArray, double[] yArray)
{
    Point3D[,] newDataArray = new Point3D[xArray.Length, yArray.Length];
    for (int i = 0; i < xArray.Length; i++)
    {
        double x = xArray[i];
        for (int j = 0; j < yArray.Length; j++)
        {
            double y = yArray[j];

```

```

        newDataArray[i, j] = new Point3D(x, y, f(x, y));
    }
}
return newDataArray;
}

private double[] CreateLinearlySpacedArray(double minValue, double maxValue,
int numberOfPoints)
{
    double[] array = new double[numberOfPoints];
    double intervalSize = (xMax - xMin) / (numberOfPoints - 1);
    for (int i = 0; i < numberOfPoints; i++)
    {
        array[i] = minValue + i * intervalSize;
    }
    return array;
}

public event PropertyChangedEventHandler PropertyChanged;

protected void RaisePropertyChanged(string property)
{
    var handler = PropertyChanged;
    if (handler != null)
    {
        handler(this, new PropertyChangedEventArgs(property));
    }
}

private Point3D[,] dataPoints;
public Point3D[,] DataPoints
{
    get { return dataPoints; }
    set
    {
        dataPoints = value;
    }
}

private double[] xAxisTicks;
public double[] XAxisTicks
{
    get { return xAxisTicks; }
    set
    {
        xAxisTicks = value;
    }
}

private double[] yAxisTicks;
public double[] YAxisTicks
{
    get { return yAxisTicks; }
    set
    {
        yAxisTicks = value;
    }
}

private double[] zAxisTicks;
public double[] ZAxisTicks
{
    get { return zAxisTicks; }
    set
    {
        zAxisTicks = value;
    }
}

```

```

    }
}

private string title;
public string Title
{
    get { return title; }
    set
    {
        title = value;
        RaisePropertyChanged("Title");
    }
}

private string xAxisLabel;
public string XAxisLabel
{
    get { return xAxisLabel; }
    set
    {
        xAxisLabel = value;
        RaisePropertyChanged("XAxisLabel");
    }
}

private string yAxisLabel;
public string YAxisLabel
{
    get { return yAxisLabel; }
    set
    {
        yAxisLabel = value;
        RaisePropertyChanged("YAxisLabel");
    }
}

private string zAxisLabel;
public string ZAxisLabel
{
    get { return zAxisLabel; }
    set
    {
        zAxisLabel = value;
        RaisePropertyChanged("ZAxisLabel");
    }
}

private double xMin, xMax, yMin, yMax;
public double[,] ColorValues { get; set; }
public ColorCoding ColorCoding { get; set; }
public Model3DGroup Lights
{
    get
    {
        var group = new Model3DGroup();
        switch (ColorCoding)
        {
            case ColorCoding.ByGradientY:
                group.Children.Add(new AmbientLight(Colors.White));
                break;
            case ColorCoding.ByLights:

```

```

        group.Children.Add(new AmbientLight(Colors.Gray));
        group.Children.Add(new PointLight(Colors.Red, new Point3D(0,
-1000, 0)));
        group.Children.Add(new PointLight(Colors.Blue, new
Point3D(0, 0, 1000)));
        group.Children.Add(new PointLight(Colors.Green, new
Point3D(1000, 1000, 0)));
        break;
    }
    return group;
}
}

public Brush SurfaceBrush
{
    get
    {
        switch (ColorCoding)
        {
            case ColorCoding.ByGradientY:
                return BrushHelper.CreateGradientBrush(Colors.Red,
Colors.White, Colors.Blue);
            case ColorCoding.ByLights:
                return Brushes.White;
        }
        return null;
    }
}

public double[,] FindGradientY(Point3D[,] data)
{
    int n = data.GetUpperBound(0) + 1;
    int m = data.GetUpperBound(1) + 1;
    var K = new double[n, m];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
        {
            var p10 = data[i + 1 < n ? i + 1 : i, j - 1 > 0 ? j - 1 : j];
            var p00 = data[i - 1 > 0 ? i - 1 : i, j - 1 > 0 ? j - 1 : j];
            var p11 = data[i + 1 < n ? i + 1 : i, j + 1 < m ? j + 1 : j];
            var p01 = data[i - 1 > 0 ? i - 1 : i, j + 1 < m ? j + 1 : j];

            double dy = p10.Y - p00.Y;
            double dz = p10.Z - p00.Z;

            K[i, j] = dz / dy;
        }
    return K;
}
}
}

```

### PlotView.xaml.cs:

```

using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Media.Media3D;

namespace WPFSurfacePlot3D
{
    public partial class SurfacePlotView : UserControl
    {

```

```

public SurfacePlotView()
{
    InitializeComponent();
    DataContext = LayoutRoot.DataContext;
    hViewport.ZoomExtentsGesture = new KeyGesture(Key.Space);
}

public Point3D[,] DataPoints
{
    get { return (Point3D[,])GetValue(DataPointsProperty); }
    set { SetValue(DataPointsProperty, value); }
}

public static readonly DependencyProperty DataPointsProperty =
DependencyProperty.Register("DataPoints", typeof(Point3D[,]),
typeof(SurfacePlotView), new
FrameworkPropertyMetadata(SurfacePlotVisual3D.SamplePoints));

public string Title
{
    get { return (string)GetValue(TitleProperty); }
    set { SetValue(TitleProperty, value); }
}

public static readonly DependencyProperty TitleProperty =
DependencyProperty.Register("Title", typeof(string), typeof(SurfacePlotView), new
FrameworkPropertyMetadata("Surface Plot Title"));

public string XAxisLabel
{
    get { return (string)GetValue(XAxisLabelProperty); }
    set { SetValue(XAxisLabelProperty, value); }
}

public static readonly DependencyProperty XAxisLabelProperty =
DependencyProperty.Register("XAxisLabel", typeof(string), typeof(SurfacePlotView),
new FrameworkPropertyMetadata("X Axis Label"));

public string YAxisLabel
{
    get { return (string)GetValue(YAxisLabelProperty); }
    set { SetValue(YAxisLabelProperty, value); }
}

public static readonly DependencyProperty YAxisLabelProperty =
DependencyProperty.Register("YAxisLabel", typeof(string), typeof(SurfacePlotView),
new FrameworkPropertyMetadata("Y Axis Label"));

public string ZAxisLabel
{
    get { return (string)GetValue(ZAxisLabelProperty); }
    set { SetValue(ZAxisLabelProperty, value); }
}

public static readonly DependencyProperty ZAxisLabelProperty =
DependencyProperty.Register("ZAxisLabel", typeof(string), typeof(SurfacePlotView),
new FrameworkPropertyMetadata("Z Axis Label"));

public bool ShowSurfaceMesh
{
    get { return (bool)GetValue(ShowSurfaceMeshProperty); }
    set { SetValue(ShowSurfaceMeshProperty, value); }
}

```

```

    public static readonly DependencyProperty ShowSurfaceMeshProperty =
DependencyProperty.Register("ShowSurfaceMesh", typeof(bool),
typeof(SurfacePlotView), new FrameworkPropertyMetadata(true));

    public bool ShowContourLines
    {
        get { return (bool)GetValue(ShowContourLinesProperty); }
        set { SetValue(ShowContourLinesProperty, value); }
    }

    public static readonly DependencyProperty ShowContourLinesProperty =
DependencyProperty.Register("ShowContourLines", typeof(bool),
typeof(SurfacePlotView), new FrameworkPropertyMetadata(true));

    public bool ShowMiniCoordinates
    {
        get { return (bool)GetValue(ShowMiniCoordinatesProperty); }
        set { SetValue(ShowMiniCoordinatesProperty, value); }
    }

    public static readonly DependencyProperty ShowMiniCoordinatesProperty =
DependencyProperty.Register("ShowMiniCoordinates", typeof(bool),
typeof(SurfacePlotView), new FrameworkPropertyMetadata(true));
}
}

```

## PlotView.xaml

```

<UserControl x:Class="WPFSurfacePlot3D.SurfacePlotView"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:h="clr-namespace:HelixToolkit.Wpf;assembly=HelixToolkit.Wpf"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:WPFSurfacePlot3D"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="300">

    <Grid x:Name="LayoutRoot">
        <Grid.RowDefinitions>
            <RowDefinition Height="1*" />
            <RowDefinition Height="11*" />
        </Grid.RowDefinitions>

        <TextBlock
            Text="{Binding Title, UpdateSourceTrigger=PropertyChanged}"
            Grid.Row="0"
            Margin="10"
            HorizontalAlignment="Center"
            />

        <h:HelixViewport3D
            x:Name="hViewport"
            Grid.Row="1"
            ZoomExtentsWhenLoaded="True"
            PanGesture="LeftClick"
            ShowViewCube="False"
            ShowCoordinateSystem="{Binding ShowMiniCoordinates,
UpdateSourceTrigger=PropertyChanged}"

```

```
        CoordinateSystemLabelX="{Binding XAxisLabel,
UpdateSourceTrigger=PropertyChanged}"
        CoordinateSystemLabelY="{Binding YAxisLabel,
UpdateSourceTrigger=PropertyChanged}"
        CoordinateSystemLabelZ="{Binding ZAxisLabel,
UpdateSourceTrigger=PropertyChanged}"
    >
    <ModelVisual3D Content="{Binding Lights,
UpdateSourceTrigger=PropertyChanged}"/>
    <local:SurfacePlotVisual3D
        DataPoints="{Binding DataPoints,
UpdateSourceTrigger=PropertyChanged}"
        ColorValues="{Binding ColorValues,
UpdateSourceTrigger=PropertyChanged}"
        SurfaceBrush="{Binding SurfaceBrush,
UpdateSourceTrigger=PropertyChanged}"
    />

    </h:HelixViewport3D>
</Grid>
</UserControl>
```