

Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра методів математичної фізики

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

**«Динамічна задача для пружного шару з жорстким
циліндричним включенням»**

**«Dynamic problem for an elastic layer with rigid
cylindrical inclusion»**

Виконав: здобувач денної форми навчання
спеціальності 113 Прикладна математика
Освітня програма «Прикладна математика»
Рогулін Артем Олександрович

Керівник: канд. фіз.-мат. наук, доц. Фесенко Г. О. _____

Рецензент: канд. фіз.-мат. наук, доц. Процеров Ю. С

Рекомендовано до захисту:

Протокол засідання кафедри

№ ____ від _____ 2023 р.

Завідувач кафедри

Захищено на засіданні ЕК № _____

Протокол № ____ від _____ 2023 р.

Оцінка _____ / _____ / _____

Голова ЕК

Одеса — 2023 р.

ЗМІСТ

Вступ	3
1 Пошук розв’язку	4
1.1 Постановка задачі та її зведення до одновимірної	4
1.2 Розв’язання векторної крайової задачі у просторі трансформант	8
2 Деталізація вихідної задачі на випадок сталих коливань, чисельні розрахунки та їх аналіз	12
Висновки	15
Список літератури	16
Додатки	17
Додаток А	17
Додаток Б	20
Додаток В	34

ВСТУП

Метою даної роботи є побудова розрахункових формул для напружень, що виникають під дією зосередженого навантаження по радіусу на нескінченний шар з жорстким циліндричним включенням, написання коду програми та аналіз напружень при різних параметрах (таких як коефіцієнт Пуассона для шару; частота коливань; відношення радіуса включення до товщини шару; відношення радіуса кола, по якому зосереджена сила, до радіуса включення). А також оцінка того, як впливають параметри на розв'язок та до яких коливань придатні знайдені розв'язки.

Огляд літератури по темі. Дефекти різної природи у пружних тілах викликають концентрацію напружень та суттєво впливають на міцність конструкції [3]. В [1] розглядаються напруження, що виникають у пружному шарі з циліндричним жорстким включенням під дією динамічного розподіленого навантаження. У [2] пропонується метод розв'язання цієї задачі, який базується на використанні векторного інтегрального перетворення типу Вебера, уникаючи використання зображень Палковича-Нейбера, що суттєво спрощує розв'язання. Дослідження, зазначені у [5], зосереджені на задачі про напруження в пружному шарі з циліндричним включенням, де враховується вага шару. Автори використали метод інтегральних перетворень для отримання числового розв'язку осесиметричної задачі теорії пружності для пружного циліндру скінченної довжини з закріпленою бічною границею та урахуванням власної ваги [6].

У [7] розглянуто динамічні напруження біля циліндричного включення різної щільності під час поширення плоскої хвилі у пружному середовищі та встановлено їх залежність від кількості хвиль падіння та коефіцієнта Пуассона, а також використання методу граничних елементів яке дозволило встановити закономірності сталих коливань плоского жорсткого включення у тривимірному пружному тілі. Робота, згадана у [8], присвячена вивченню рівноваги пружного скінченного циліндра, який піддається осесиметричному нормальному навантаженню.

РОЗДІЛ 1

ПОШУК РОЗВ'ЯЗКУ

Був розроблений точний розв'язок для нестационарної задачі пружного шару з циліндричним жорстким включенням, де на циліндричній поверхні задані умови гладкого контакту. Одна з граней шару піддана осесиметричному нестационарному стискаючому навантаженню, яке зосереджене по колу радіуса b . Інша грань має умови сполучення з абсолютно жорстким основою або опирається на гладку основу без тертя. Для вирішення задачі з побудовою поля переміщень та напружень в шарі були послідовно застосовані інтегральні перетворення Лапласа та Вебера, що призводить до векторної неоднорідної крайової задачі щодо невідомих трансформант переміщень. Задача була вирішена за допомогою матричного диференціального числення. Після застосування зворотних інтегральних перетворень, було отримано хвильове поле шару. Були досліджені нормальні напруження на циліндричній поверхні включення та на нижній грані пружного шару. Розв'язок задачі був розглянутий для випадку квазістатичних коливань.

1.1 Постановка задачі та її зведення до одновимірної

Розглянемо нескінченний пружний шар з циліндричним жорстким включенням радіуса a :

$$a \leq r < \infty, \quad -\pi \leq \phi \leq \pi, \quad 0 \leq z \leq h$$

На грані $z = h$ діє осесиметричне нормальне нестационарне стискаюче навантаження

$$\sigma_z(r, h, t) = -p(r, t), \quad \tau_{zr}(r, h, t) = 0 \quad (1.1)$$

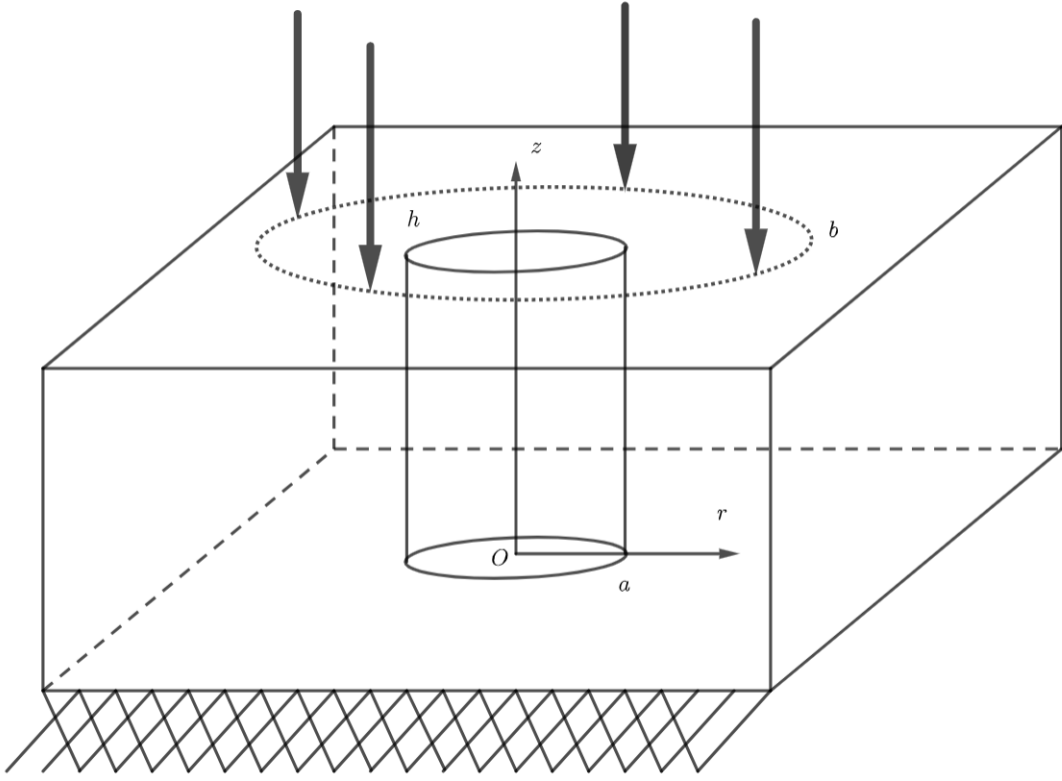


Рис. 1.1.

На грані $z = 0$ задано умови жорсткого закріплення

$$u_z(r, 0, t) = 0, \quad u_r(r, 0, t) = 0 \quad (1.2)$$

В області контакту жорсткого включення з шаром задано умови ідеального контакту

$$u_r(a, z, t) = 0, \quad \tau_{rz}(a, z, t) = 0 \quad (1.3)$$

Необхідно побудувати хвильове поле пружного шару $u_r(r, z, t) = u(r, z, t)$, $u_z(r, z, t) = w(r, z, t)$, що задовольняє осесиметричні рівняння руху при нульових початкових умовах

$$\begin{aligned} r^{-1} \frac{\partial}{\partial r} \left[r \frac{\partial u}{\partial r} \right] - r^{-2} u(r, z, t) + \frac{\kappa - 1}{\kappa + 1} \frac{\partial^2 u(r, z, t)}{\partial z^2} + \frac{2}{\kappa + 1} \frac{\partial^2 w(r, z, t)}{\partial r \partial z} = \\ = \frac{\kappa - 1}{\kappa + 1} \frac{\rho}{G} \frac{\partial^2 u(r, z, t)}{\partial t^2} \end{aligned}$$

$$r^{-1} \frac{\partial}{\partial r} \left[r \frac{\partial w}{\partial r} \right] + \frac{\kappa + 1}{\kappa - 1} \frac{\partial^2 w(r, z, t)}{\partial z^2} + \frac{2}{\kappa - 1} r^{-1} \frac{\partial}{\partial r} \left[r \frac{\partial u(r, z, t)}{\partial z} \right] = \frac{\rho}{G} \frac{\partial^2 w(r, z, t)}{\partial t^2} \quad (1.4)$$

де $\kappa = 3 - 4 \cdot \mu$, μ - коефіцієнт Пуассона, G - модуль зсуву, ρ - густина.

Зауважимо, що $c^2 = \frac{G}{\rho}$ - швидкість розповсюдження поперечної хвилі,

$c_1^2 = \frac{\kappa + 1}{\kappa - 1} c^2$ - швидкість розповсюдження поздовжньої хвилі та введемо наступні безрозмірні координати

$$\rho = a^{-1}r, \quad \xi = h^{-1}z, \quad \tau = a^{-1}ct, \quad \rho \in [1, \infty), \quad \xi \in [0, 1]. \quad (1.5)$$

Враховуючи заміну змінних (1.5), позначимо переміщення $u(r, z, t) = U(\rho, \xi, \tau)$, $w(r, z, t) = W(\rho, \xi, \tau)$ тоді система рівнянь руху (1.4) у нових координатах набуває вигляду

$$\begin{aligned} \rho^{-1} \frac{\partial}{\partial \rho} \left[\rho \frac{\partial U}{\partial \rho} \right] - \rho^{-2} U + \alpha^2 \frac{\kappa - 1}{\kappa + 1} \frac{\partial^2 U}{\partial \xi^2} + \alpha^2 \frac{2}{\kappa + 1} \frac{\partial^2 W}{\partial \rho \partial \xi} &= \frac{\kappa - 1}{\kappa + 1} \frac{\partial^2 U(\rho, \xi, \tau)}{\partial \tau^2} \\ \rho^{-1} \frac{\partial}{\partial \rho} \left[\rho \frac{\partial W}{\partial \rho} \right] + \alpha^2 \frac{\kappa + 1}{\kappa - 1} \frac{\partial^2 W}{\partial \xi^2} + \alpha \frac{2}{\kappa - 1} \rho^{-1} \frac{\partial}{\partial \rho} \left[\rho \frac{\partial U}{\partial \xi} \right] &= \frac{\partial^2 W(\rho, \xi, \tau)}{\partial \tau^2} \end{aligned} \quad (1.6)$$

Умови гладкого контакту (1.3) для $r = a$, враховуючі заміну (1.5), записано у вигляді

$$U(1, \xi, \tau) = 0, \quad \frac{\partial W(1, \xi, \tau)}{\partial \rho} = 0 \quad (1.7)$$

Аналогічно, крайові умови (1.2) на грані $z = 0$ та (1.1) на грані $z = h$, запишемо наступним чином

$$U(\rho, 0, \tau) = 0, \quad W(\rho, 0, \tau) = 0 \quad (1.8)$$

$$\frac{\partial W(\rho, 1, \tau)}{\partial \rho} + \alpha \frac{\partial U(\rho, 1, \tau)}{\partial \xi} = 0$$

$$\frac{\partial U(\rho, 1, \tau)}{\partial \rho} + \rho^{-1} U(\rho, 1, \tau) + \bar{\mu} \alpha \frac{\partial W(\rho, 1, \tau)}{\partial \xi} = -\frac{a \kappa - 1}{G 3 - \kappa} p(\rho, \tau) \quad (1.9)$$

$$\text{де } \bar{\mu} = \frac{1-\mu}{\mu} = \frac{1+\kappa}{3-\kappa}, \quad \alpha = \frac{a}{h}$$

Останню умову у (1.9) отримано з граничної умови (1.1) $\sigma_z(r, h, t) = -p(r, t)$ за формулою $\sigma_z = (\lambda + 2G)\frac{\partial w}{\partial z} + \lambda\left(\frac{\partial u}{\partial r} + \frac{1}{r}u\right)$ (λ, G - сталі Ламе, $\lambda = \frac{2\mu}{1-2\mu}G = \frac{3-\kappa}{\kappa-1}G$), тобто $\sigma_z = G\frac{1+\kappa}{\kappa-1}\frac{\partial w}{\partial z} + G\frac{3-\kappa}{\kappa-1}\left(\frac{\partial u}{\partial r} + \frac{1}{r}u\right)$, з урахуванням заміни (1.5). Послідовно застосуємо до системи (1.6) та граничних умов (1.8), (1.9) інтегральне перетворення Лапласа за змінною τ

$$f_p(\rho, \xi) = \int_0^\infty f(\rho, \xi, \tau)e^{-p\tau}d\tau, \quad f(\rho, \xi, \tau) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} f_p(\rho, \xi)e^{p\tau}dp \quad (1.10)$$

та перетворення Вебера за змінною ρ

$$U_{p\lambda}(\xi) = \int_1^\infty \rho U_p(\rho, \xi) \chi_1(\rho, \lambda) d\rho, \quad U_p(\rho, \xi) = \int_0^\infty \frac{\lambda U_{p\lambda}(\xi) \chi_1(\rho, \lambda) d\lambda}{J_1^2(\lambda) + N_1^2(\lambda)}$$

$$W_{p\lambda}(\xi) = \int_1^\infty \rho W_p(\rho, \xi) \chi_0(\rho, \lambda) d\rho, \quad W_p(\rho, \xi) = \int_0^\infty \frac{\lambda W_{p\lambda}(\xi) \chi_0(\rho, \lambda) d\lambda}{J_1^2(\lambda) + N_1^2(\lambda)} \quad (1.11)$$

де

$$\chi_i(\rho, \lambda) = J_i(\rho\lambda)N_i(\lambda) - N_i(\rho\lambda)J_1(\lambda), \quad i = 0, 1$$

$J_i(\lambda)$ - функції Бесселя, $N_i(\lambda)$ - функції Неймана, $i = 0, 1$. Під час застосування інтегрального перетворення Вебера граничні умови (1.7) автоматично реалізовано. Це приводить до системи звичайних диференціальних рівнянь у просторі трансформант

$$U_{p\lambda}''(\xi) - \frac{2}{\kappa-1}\lambda_* W_{p\lambda}'(\xi) - \frac{\kappa+1}{\kappa-1}\lambda_*^2 U_{p\lambda}(\xi) - p^2 \alpha^{-2} U_{p\lambda}(\xi) = 0 \quad (1.12)$$

$$W_{p\lambda}''(\xi) + \frac{2}{\kappa+1}\lambda_* U_{p\lambda}'(\xi) - \frac{\kappa-1}{\kappa+1}\lambda_*^2 W_{p\lambda}(\xi) - p^2 \alpha^{-2} \frac{\kappa-1}{\kappa+1} W_{p\lambda}(\xi) = 0, \quad 0 < \xi < 1$$

де $\lambda_* = \lambda\alpha^{-1}$. Крайові умови (1.8), (1.9) у трансформантах (1.10), (1.11)

записано у формі

$$U_{p\lambda}(0) = 0, \quad W_{p\lambda}(0) = 0 \quad (1.13)$$

$$\lambda_* U_{p\lambda}(1) + \bar{\mu} W'_{p\lambda}(1) = -A_* p_{p\lambda}, \quad U'_{p\lambda}(1) - \lambda_* W_{p\lambda}(1) = 0$$

$$\begin{aligned} A_* &= a(G\alpha)^{-1} \frac{\kappa - 1}{3 - \kappa}, \\ p_{p\lambda} &= \int_1^\infty \rho p_p(a\rho) \chi_0(\rho, \lambda) d\rho, \\ p_p(a\rho) &= \int_0^\infty p(a\rho, \frac{a}{c}\tau) e^{-p\tau} d\tau \end{aligned} \quad (1.14)$$

Отриману одновимірну крайову задачу (1.12), (1.13) переформулюємо у векторній формі.

1.2 Розв'язання векторної крайової задачі у просторі трансформант

Введемо до розгляду невідомий вектор трансформант переміщень, матриці та диференціальний оператор

$$\begin{aligned} \mathbf{y}(z) &= \begin{pmatrix} U_{p\lambda}(\xi) \\ W_{p\lambda}(\xi) \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 0 & -\frac{1}{\kappa - 1} \\ \frac{1}{\kappa + 1} & 0 \end{pmatrix}, \\ \mathbf{P} &= \begin{pmatrix} \frac{\kappa + 1}{\kappa - 1} & 0 \\ 0 & \frac{\kappa - 1}{\kappa + 1} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{\kappa - 1}{\kappa + 1} \end{pmatrix}, \\ \mathbf{I}_1 &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{I}_2 = \begin{pmatrix} 0 & \bar{\mu} \\ 1 & 0 \end{pmatrix}, \quad \mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

$$L_2 \mathbf{y}(\xi) = \mathbf{I} \cdot \mathbf{y}''(\xi) 2\lambda_* \mathbf{Q} \cdot \mathbf{y}'(\xi) - \lambda_*^2 \mathbf{P} \cdot \mathbf{y}(\xi) - p^2 \alpha^{-2} \mathbf{T} \cdot \mathbf{y}(\xi) \quad (1.15)$$

Крайові умови (1.13) переформульовано за допомогою граничних функціоналів

$$U_0[\mathbf{y}(\xi)] = \mathbf{I} \cdot \mathbf{y}(0), \quad U_1[\mathbf{y}(\xi)] = \lambda_* \mathbf{I}_1 \cdot \mathbf{y}(1) + \mathbf{I}_2 \cdot \mathbf{y}'(1) \quad (1.16)$$

Враховуючі (1.15), (1.16), задачу (1.12), (1.13) запишемо у векторній форм

$$\begin{cases} L_2 \mathbf{y}(\xi) = \mathbf{0}, & 0 < \xi < 1 \\ U_i[\mathbf{y}(\xi)] = \Upsilon_i, & i = 0, 1 \end{cases} \quad (1.17)$$

де $\Upsilon_0 = (0, 0)^T$, $\Upsilon_1 = (-A_* p_{p\lambda}, 0)^T$. Як показано у [4] розв'язок задачі (1.17) можна сконструювати у зображенні

$$\mathbf{y}(\xi) = \Psi_0(\xi) + \Psi_1(\xi) \cdot \Upsilon_1 \quad (1.18)$$

де $\Psi_i(\xi), i = 0, 1$ - матрична базисна система розв'язків. Під матричною базисною системою розв'язків вважається така система, що задовольняє задачі

$$\begin{cases} L_2 \Psi_i = \mathbf{0} & 0 < \xi < 1 \\ U_j[\Psi_i(\xi)] = \delta_{ij}, & i, j = 0, 1 \end{cases}$$

де диференціальний оператор та граничні функціонали визначено (1.15), (1.16), δ_{ij} - символ Кронекера. Методика побудови $\Psi_i(\xi), i = 0, 1$ викладена у [4]. Відповідно до цієї методики спочатку отримано загальний розв'язок однорідного рівняння з (1.17), для чого розв'язано однорідне матричне рівняння

$$L_2 \mathbf{Y}(\xi) = \mathbf{0}, \quad 0 < \xi < 1 \quad (1.19)$$

де $\mathbf{Y}(\xi)$ - невідома матриця порядку 2×2 . Розв'язок цього рівняння вибрано у формі $\mathbf{Y}(\xi) = e^{\eta\xi} \cdot \mathbf{I}$, що привело до співвідношення $L_2 e^{\eta\xi} \cdot \mathbf{I} = \mathbf{M}(\eta) \cdot e^{\eta\xi}$, звідки маємо

$$\mathbf{Y}(\xi) = \frac{1}{2\pi i} \int_C \mathbf{M}^{-1}(\eta) e^{\eta\xi} d\eta,$$

тут C - замкнений контур, що містить всі полюси підінтегрального виразу. Цими полюсами будуть нулі визначника матриці $\mathbf{M}^{-1}(\eta)$, що має вигляд

$$\mathbf{M}^{-1}(\eta) = \frac{\mathbf{M}_0(\eta)}{\det \mathbf{M}(\eta)}$$

$$\det \mathbf{M}(\eta) = \left[\eta^2 - \left(\sqrt{\lambda^2 + p^2 \alpha^{-2}} \right)^2 \right] \left[\eta^2 - \left(\sqrt{\lambda_*^2 + \frac{\kappa - 1}{\kappa + 1} p^2 \alpha^{-2}} \right)^2 \right]$$

$$\mathbf{M}_0(\eta) = \begin{pmatrix} \eta^2 - \lambda_*^2 \frac{\kappa - 1}{\kappa + 1} - p^2 \alpha^{-2} \frac{\kappa - 1}{\kappa + 1} & \frac{2}{\kappa - 1} \lambda_* \eta \\ -\frac{2}{\kappa + 1} \lambda_* \eta & \eta^2 - \lambda_*^2 \frac{\kappa + 1}{\kappa - 1} - p^2 \alpha^{-2} \end{pmatrix}.$$

Позначимо розв'язки, що відповідають різним полюсам, як

$$\mathbf{Y}_{\mp} = \frac{1}{2\pi i} \int_C \frac{e^{\eta \xi} \mathbf{M}_0(\eta) d\eta}{\det \mathbf{M}(\eta)}.$$

Якщо зафіксувати полюси $\eta = \sqrt{\lambda_*^2 + p^2 \alpha^{-2}}$, $\eta = \sqrt{\lambda_*^2 + \frac{\kappa - 1}{\kappa + 1} p^2 \alpha^{-2}}$ та контур C_+ , що їх містить, отримаємо розв'язок $\mathbf{Y}_+(\xi)$, що зростає, якщо $\xi \rightarrow \infty$; при фіксації полюсів $\eta = -\sqrt{\lambda_*^2 + p^2 \alpha^{-2}}$, $\eta = -\sqrt{\lambda_*^2 + \frac{\kappa - 1}{\kappa + 1} p^2 \alpha^{-2}}$ та контуру C_- отримаємо спадаючий розв'язок $\mathbf{Y}_-(\xi)$. Використання теореми про лишки завершує розв'язання рівняння (1.19)

$$\mathbf{Y}_{\mp}(\xi) = \frac{\alpha}{2p^2} \left[e^{-\frac{1}{\alpha} \Delta_0 \cdot \xi} \begin{pmatrix} \mp \Delta_0 & \frac{\kappa + 1}{\kappa - 1} \\ -\lambda & \pm \frac{\kappa + 1}{\kappa - 1} \frac{\lambda^2}{\Delta_0} \end{pmatrix} + e^{-\frac{1}{\alpha} \Delta_1 \cdot \xi} \begin{pmatrix} \pm \frac{\lambda^2}{\Delta_1} & -\frac{\kappa + 1}{\kappa - 1} \lambda \\ \lambda & \mp \frac{\kappa + 1}{\kappa - 1} \Delta_1 \end{pmatrix} \right]$$

$$\Delta_0 = \sqrt{\lambda^2 + p^2}, \quad \Delta_1 = \sqrt{\lambda^2 + \frac{\kappa - 1}{\kappa + 1} p^2}$$

Побудовано базисні матриці

$$\Psi_j(\xi) = \mathbf{Y}_-(\xi) \mathbf{C}_j^0 + \mathbf{Y}_+(\xi) \mathbf{C}_j^1, \quad j = 0, 1$$

де C_j^i , $i, j = 0, 1$ - матриці сталих порядку 2×2 .

$$\mathbf{C}_1^1 = (U_1[\mathbf{Y}_+] - U_1[\mathbf{Y}_-] \cdot (U_0[\mathbf{Y}_-])^{-1} \cdot U_0[\mathbf{Y}_+])^{-1}, \quad \mathbf{C}_1^0 = -(U_0[\mathbf{Y}_-])^{-1} \cdot U_0[\mathbf{Y}_+] \cdot \mathbf{C}_1^1.$$

де $U_i[\mathbf{Y}_\pm]$, $i = 0, 1$ - значення спадаючого та зростаючого розв'язків однорідного рівняння (1.19) на граничних функціоналах, визначених у (1.16). За рахунок задоволення граничних умов, отримано, що $C_1^0 = C_1^1 = C$ та вигляд базисної матриці спрощується

$$\Psi_1(\xi) = (\mathbf{Y}_-(\xi) + \mathbf{Y}_+(\xi)) \cdot \mathbf{C}.$$

Остаточно маємо

$$\Psi_1(\xi) = \frac{\alpha}{\det \Psi} \begin{pmatrix} \frac{1}{2} \frac{3-\kappa}{\kappa-1} \lambda \Psi_{11} & -\Psi_{12} \\ \frac{1}{2} \frac{3-\kappa}{\kappa-1} \Psi_{21} & \lambda \Psi_{22} \end{pmatrix} \quad (1.20)$$

$$\begin{aligned} \det \Psi &= 2\lambda^2(2\lambda^2 + p^2) - (4\lambda^4 + 2\lambda^2 p^2 + \frac{1}{2}p^4) \cosh \frac{\Delta_0}{\alpha} \cdot \cosh \frac{\Delta_1}{\alpha} + \\ &+ \frac{\lambda^2 \cdot (4\lambda^4 + \frac{2(3\kappa+1)}{\kappa+1} \lambda^2 p^2 + \frac{\kappa-3}{2(\kappa+1)} p^4)}{\Delta_0 \cdot \Delta_1} \cdot \sinh \frac{\Delta_0}{\alpha} \cdot \sinh \frac{\Delta_1}{\alpha} \end{aligned}$$

Запишемо розв'язок (1.18) векторної крайової задачі (1.17)

$$\mathbf{y}(z) = \begin{pmatrix} U_{p\lambda}(\xi) \\ W_{p\lambda}(\xi) \end{pmatrix} = -A_* \cdot p_{p\lambda} \cdot \frac{\alpha}{\det \Psi} \cdot \begin{pmatrix} \frac{1}{2} \frac{3-\kappa}{\kappa-1} \lambda \Psi_{11} \\ -\Psi_{12} \end{pmatrix}$$

Застосувавши обернене перетворення Лапласа змінної p і обернене перетворення Вебера змінної λ , отримуємо поле хвиль для шару.

$$\begin{pmatrix} U(\rho, \xi, \tau) \\ W(\rho, \xi, \tau) \end{pmatrix} = \frac{1}{2\pi} \int_{\gamma-i\infty}^{\gamma+i\infty} \int_0^\infty \frac{\lambda}{J_1^2(\lambda) + N_1^2(\lambda)} \cdot \begin{pmatrix} U_{p\lambda}(\xi) \cdot \chi_1(\rho, \lambda) \\ W_{p\lambda}(\xi) \cdot \chi_0(\rho, \lambda) \end{pmatrix} e^{p\tau} dp d\lambda \quad (1.21)$$

Таким чином, метод, запропонований у роботі [1], дав змогу отримати точний розв'язок для нестационарної задачі (1.1-1.4) за допомогою просторових інтегральних перетворень (1.10, 1.11).

РОЗДІЛ 2

ДЕТАЛІЗАЦІЯ ВИХІДНОЇ ЗАДАЧИ НА ВИПАДОК СТАЛИХ КОЛИВАНЬ, ЧИСЕЛЬНІ РОЗРАХУНКИ ТА ЇХ АНАЛІЗ

Деталізація відповідає постановці задачі для випадку квазістатичного навантаження, заданого на верхній грані пружного шару. За рахунок аналітичного подання хвильового поля (1.21) у просторі трансформант Лапласа покладемо $p = -\delta(r-b)$, де δ - дельта функція Дірака; ω - частота коливань, у всіх попередніх формулах. Таким чином, отримано наступні нормальні напруження на нижній грані пружного шару:

$$\begin{aligned} \sigma_{\xi}(\rho, \xi = 0; \omega) &= \frac{b}{a} \int_0^{\omega-0} \frac{\omega^2 \chi_0(\frac{b}{a}, \lambda) \cdot \chi_0(\rho, \lambda)}{J_1^2(\lambda) + N_1^2(\lambda)} \cdot \lambda \cdot \Phi_1(\lambda; \omega) d\lambda + \\ &+ \frac{b}{a} \int_{\omega+0}^{\infty} \frac{\omega^2 \chi_0(\frac{b}{a}, \lambda) \cdot \chi_0(\rho, \lambda)}{J_1^2(\lambda) + N_1^2(\lambda)} \cdot \lambda \cdot \Phi_2(\lambda; \omega) d\lambda, \\ \rho &\in [1, \infty). \end{aligned}$$

та на циліндричній поверхні:

$$\begin{aligned} \sigma_{\rho}(\rho = 1, \xi; \omega) &= \\ &= \int_0^{\omega-0} \frac{\chi_0(\frac{b}{a}, \lambda) \cdot \lambda}{J_1^2(\lambda) + N_1^2(\lambda)} \cdot [2\lambda^2 \cdot \chi_1(1, \lambda) \cdot \Phi_3(\lambda, \xi; \omega) - \chi_0(1, \lambda) \cdot \Phi_4(\lambda, \xi; \omega)] d\lambda + \\ &+ \int_{\omega+0}^{\infty} \frac{\chi_0(\frac{b}{a}, \lambda) \cdot \lambda}{J_1^2(\lambda) + N_1^2(\lambda)} \cdot [2\lambda^2 \cdot \chi_1(1, \lambda) \cdot \Phi_5(\lambda, \xi; \omega) - \chi_0(1, \lambda) \Phi_6(\lambda, \xi; \omega)] d\lambda, \\ \xi &\in [0, 1] \end{aligned}$$

які реалізовані в коді у додатку А. Формули підінтегральних функції $\Phi_i, i = \overline{1,6}$ на випадок жорсткого зачіплення на нижній грані шару подані у додатку Б, де також знаходиться код з їх реалізацією.

В нас є декілька параметрів на які ми можемо змінювати, це:

ω - частота коливань;

$\frac{b}{a}$ - відношення радіуса кола, по якому зосереджена сила, до радіуса включення ($\frac{b}{a} > 1$);

$\alpha = \frac{a}{h}$ - відношення радіуса включення до товщини шару;

$\kappa = 3 - 4\mu$, де μ - коефіцієнт Пуассона.

Спочатку розглянемо функції підінтегральні функції для σ_ξ з різними параметрами. Наприклад для $\frac{b}{a} = 2$; $\mu = 1/4$; $\alpha = 0.5$ та різних ω :

Як можемо бачити, для частот коливань 0.79 та більше, в підінтегральній

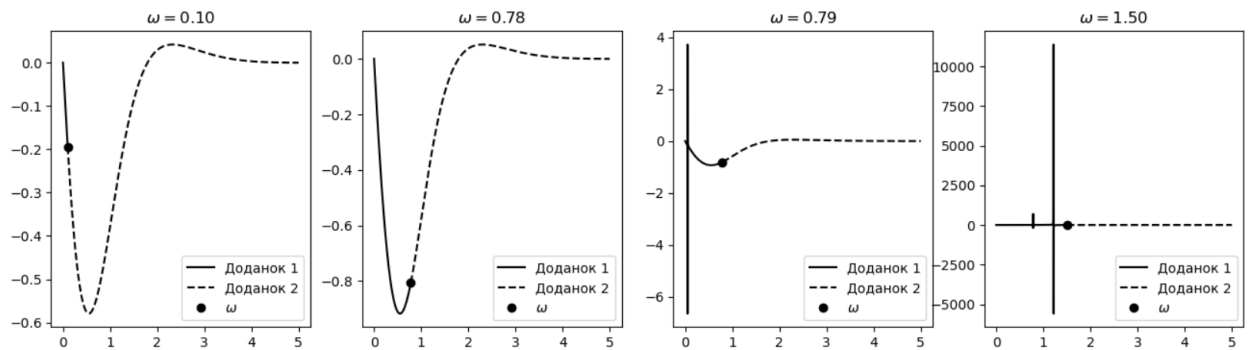


Рис. 2.1.

функції з'являються розриви. Але, якщо збільшити $\alpha = 1.0$ то отримаємо що розриви отримуються для більших частот, тобто ми можемо використовувати нашу функцію до певної сили коливань:

Теж саме працює і з σ_ρ , графіки можна переглянути у додатку В. Значення

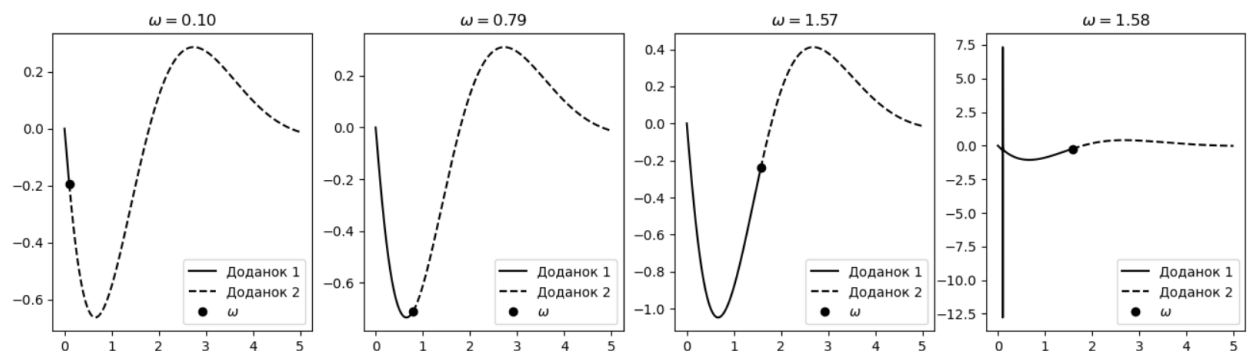
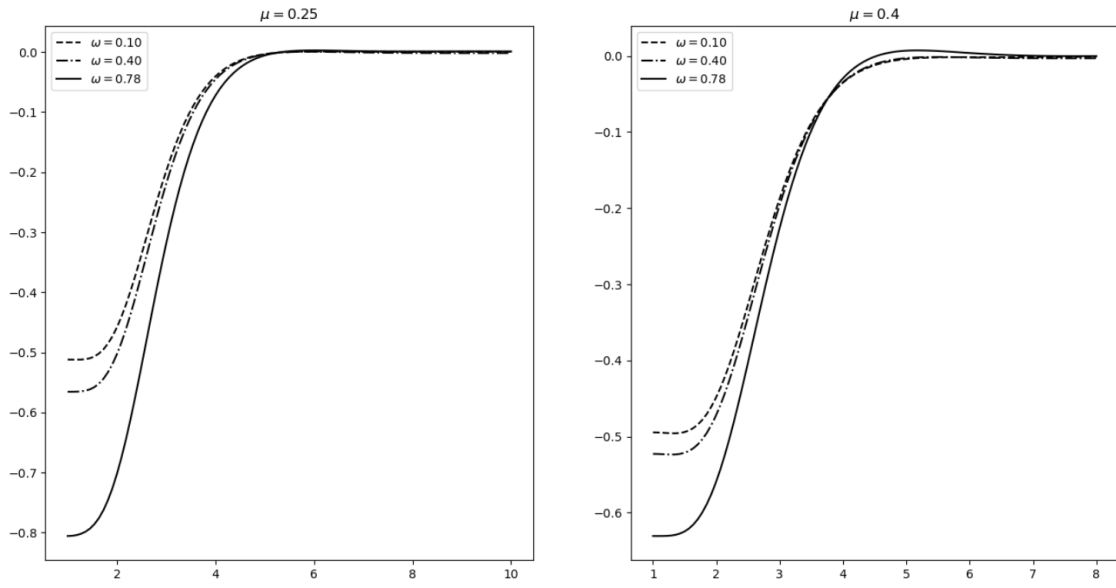
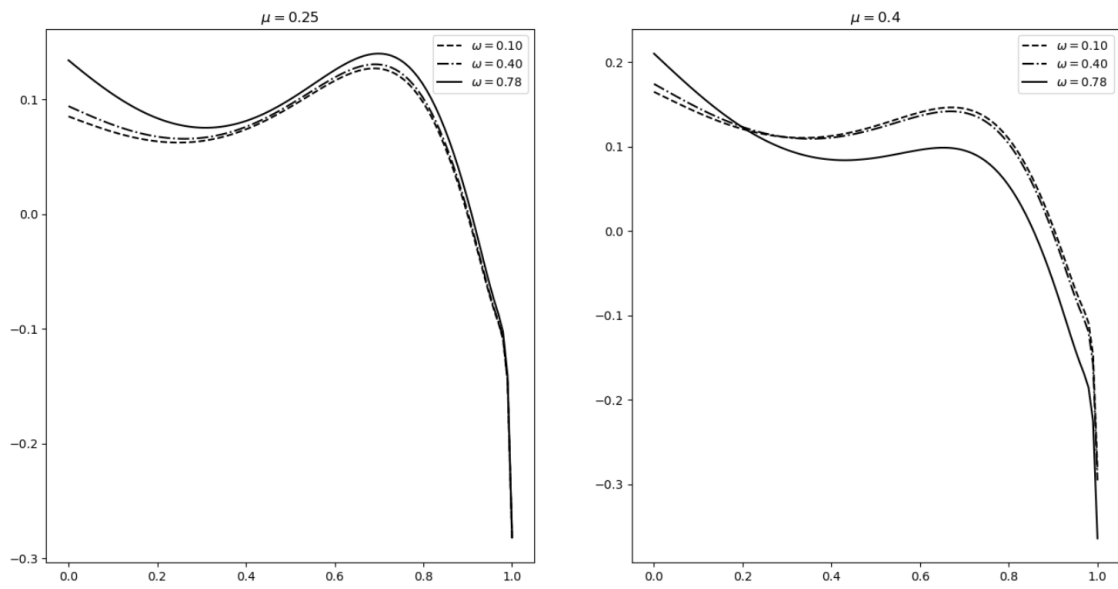


Рис. 2.2.

брали для $\rho = 1$. μ же не впливає на те, до якої частоти ми можемо використовувати ці формули. Це також можна переглянути у додатку В. Отже ми можемо мати достовірні σ_ξ та σ_ρ лише для певних ω .

Рис. 2.3. σ_ξ для $\alpha = 0.5$ Рис. 2.4. σ_ρ для $\alpha = 0.5$

ВИСНОВКИ

Отримано точний розв'язок нестационарної задачі для нескінченного пружного шару з циліндричним жорстким включенням (на поверхні якого задані умови гладкого контакту).

Досліджені нормальні напруження виникаючі на циліндричній поверхні та на нижній грані шару під дією нормального осесиметричного навантаження по колу радіусу b .

Оцінена здатність знайдених функцій вимірювати коливання (до якої сили коливання можуть бути відображені знайденою формулою) та взаємозв'язок між силою коливань, частотою коливань (ω) та відношенням радіусу включення до товщини шару (α).

Запропонований підхід можна застосувати для пошуку та оцінки розв'язків інших задач динамічного типу теорії пружності. Також спираючись на цю роботу, можна буде відшукати формули для більших коливань.

СПИСОК ЛІТЕРАТУРИ

1. Fesenko, A.A., Moyseenok, A.P. Exact Solution of a Nonstationary Problem for the Elastic Layer with Rigid Cylindrical Inclusion. J Math Sci. 249, 478-495 (2020). <https://doi.org/10.1007/s10958-020-04954-3>
2. Попов Г. Я. Точные решения некоторых краевых задач механики деформируемого твердого тела. Осесимметричное контактное взаимодействие бесконечной упругой плиты с абсолютно жестким включением / Вайсфельд Н. Д., Реут В. В. / Одесса: Астропринт. - 2013. - 424 с.
3. Попов Г. Я. Концентрация упругих напряжений возле штампов, разрезв, тонких включений и подкреплений. Москва: Наука. 1982. - 342 с.
4. Попов Г. Я., Абдыманапов С. А., Ефимов В. В. Функции и матрицы Грина одномерных краевых задач. - Алматы: Изд. Рауан. 1999. - 133 с.
5. Попов Г. Я., Вайсфельд Н. Д. Осесимметричная задача теории упругости для бесконечной плиты с цилиндрическим включением при учете ее удельного веса // Прикл. механика. - 2014. - Т. 50, № 6. - С. 27-38.
6. Попов Г. Я., Процеров Ю. С. Осесимметричная задача для упругого цилиндра конечной длины с зашцеиленной боковой поверхностью с учетом собственного веса // Мат. методи та фіз.-мех. поля. - 2014. - 57, № 1. - С. 57-68.
7. Kit H. S., Mykhas'skiv V. V., Khaj O. M. Analysis of the steady oscillations of a plane absolutely rigid inclusion in a three-dimensional elastic body by the boundary element method // J. Appl. Math. and Mech. - 2002. - 66 (5). P. 817-824
8. Meleshko V. V., Tokovyy Yu.V. Equilibrium of an elastic finite cylinder under axisymmetric discontinuous normal loadings // Journal of Engineering Mathematics. - 2013. - 78(1). P. 143-166. (doi: 10.1007/s10665- 011-9524-y)

ДОДАТКИ

Додаток А. Код для обчислення напружень

σ_ξ та σ_ρ

Тут указані пакети для роботи коду, а також реалізовані функції $\chi_i, i = 0, 1$, чисельне інтегрування методом Сімпсона та нормальне напруження на нижній грані шару й нормальне напруження на циліндричній поверхні жорсткого включення.

```
# Пакети для роботи коду
import numpy as np
import scipy as sc
import matplotlib.pyplot as plt
import cmath

from scipy.special import j1, j0, y1, y0

# Функції  $\chi_i$  з індексом 0 та 1
def chi_0(rho, lamb):
    return j0(rho * lamb) * y1(lamb) - j1(lamb) * y0(rho * lamb)

def chi_1(rho, lamb):
    return j1(rho * lamb) * y1(lamb) - j1(lamb) * y1(rho * lamb)

# Функція яка повертає значення підантегральної функції
# доданку з  $\sigma_{\xi}$ 
def sigma_xi_part(rho, params, lamb, Phi_func):
    omega, b, a, alpha, kappa, xi = params
    return ((omega**2 * chi_0(b / a, lamb) * chi_0(rho, lamb))
            / (j1(lamb)**2 + y1(lamb)**2) * lamb
            * Phi_func(lamb, params))

# Функція яка обчислює значення  $\sigma_{\xi}$ 
```

```

def sigma_xi(rho, omega, b, a, alpha, kappa, xi, Phi):
    first_Phi, second_Phi = Phi
    params = (omega, b, a, alpha, kappa, xi)
    # Precision params
    steps_num = 10000
    omega_dist = 0.00000000001
    lam1_start = 0.00000000001
    lam2_end = 50
    # lamb < omega
    lam1 = np.linspace(lam1_start, omega, steps_num,
                       endpoint=False)
    term1 = integrate(lam1, rho, params, sigma_xi_part,
                      first_Phi)

    #lamb > omega
    lam2 = np.linspace(lam2_end, omega, steps_num,
                       endpoint=False)[::-1]
    term2 = integrate(lam2, rho, params, sigma_xi_part,
                      second_Phi)
    return (b / a) * (term1 + term2)

def sigma_rho_part(xi, params, lamb, phi_funcs):
    first_func, second_func = phi_funcs
    omega, b, a, alpha, kappa, rho = params
    numerator = chi_0(b/a, lamb) * lamb
    denominator = (j1(lamb)**2 + y1(lamb)**2)
    term1 = (2 * lamb**2 * chi_1(1, lamb)
             * first_func(lamb, xi, params))
    term2 = chi_0(1, lamb) * second_func(lamb, xi, params)
    result = numerator / denominator * (term1 - term2)
    return result

def sigma_rho(xi, omega, b, a, alpha, kappa, rho, Phi):

```

```

Phi3, Phi4, Phi5, Phi6 = Phi
params = (omega, b, a, alpha, kappa, rho)
# Precision params
steps_num = 10000
omega_dist = 0.0000000001
lam1_start = 0.0000000001
lam2_end = 50
# lamb < omega
lam1 = np.linspace(lam1_start, omega, steps_num,
                    endpoint=False)
term1 = integrate(lam1, xi, params, sigma_rho_part,
                  (Phi3, Phi4))

#lamb > omega
lam2 = np.linspace(lam2_end, omega, steps_num,
                    endpoint=False)[::-1]
term2 = integrate(lam2, xi, params, sigma_rho_part,
                  (Phi5, Phi6))

return term1 + term2

# Метод Сімпсона
# (передаємо параметри та функції які хочемо обчислити)
def integrate(interval, x, params, sigma_func, phi_funcs):
    prev_lamb = interval[0]
    res = 0
    for lamb in interval[1::]:
        dist = lamb - prev_lamb
        start = sigma_func(x, params, prev_lamb, phi_funcs)
        mid = sigma_func(x, params, (prev_lamb + lamb) / 2,
                          phi_funcs)
        end = sigma_func(x, params, lamb, phi_funcs)
        res += dist * (start + 4 * mid + end) / 6
        prev_lamb = lamb
    return res

```

Додаток Б. Код та функції для обчислення $\Phi_i^{(0)}, i = \overline{1, 6}$

$$\Phi_1^{(0)}(\lambda; \omega) = \frac{F_1^*(\lambda; \omega)}{\det \Psi_1^*}, \quad (\lambda < \omega)$$

$$F_1^*(\lambda; \omega) = \left(\frac{1}{2}\omega^2 - \lambda^2\right) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right) + \\ + \lambda^2 \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right)$$

$$\det \Psi_1^* = 2\lambda^2(2\lambda^2 - \omega^2) - (4\lambda^4 - 2\lambda^2\omega^2 + \frac{1}{2}\omega^4) \cdot \\ \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right) + \\ + \frac{\lambda^2 \cdot \left(4\lambda^4 - \frac{2(3\kappa+1)}{\kappa+1}\lambda^2\omega^2 + \frac{5\kappa-3}{2(\kappa+1)}\omega^4\right)}{\omega^2 \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2}} \cdot \\ \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right) \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right)$$

$$\Phi_2^{(0)}(\lambda; \omega) = \frac{F_1(\lambda; \omega)}{\det \Psi_1}, \quad (\lambda > \omega)$$

$$F_1(\lambda; \omega) = \left(\frac{1}{2}\omega^2 - \lambda^2\right) \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \right) + \lambda^2 \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa-1}{\kappa+1}\omega^2} \right)$$

$$\begin{aligned}
\det \Psi_1 = & 2\lambda^2(2\lambda^2 - \omega^2) - (4\lambda^4 - 2\lambda^2\omega^2 + \frac{1}{2}\omega^4) \cdot \\
& \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2}\right) \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa-1}{\kappa+1}\omega^2}\right) + \\
& + \frac{\lambda^2 \cdot (4\lambda^4 - \frac{2(3\kappa+1)}{\kappa+1}\lambda^2\omega^2 + \frac{5\kappa-3}{2(\kappa+1)}\omega^4)}{\sqrt{\lambda^2 - \omega^2} \cdot \sqrt{\lambda^2 - \frac{\kappa-1}{\kappa+1}\omega^2}} \cdot \\
& \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2}\right) \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa-1}{\kappa+1}\omega^2}\right)
\end{aligned}$$

$$\Phi_3^{(0)}(\lambda, \xi; \omega) = \frac{F_3^*(\lambda, \xi; \omega)}{\det \Psi_1^*}, \quad (\lambda < \omega)$$

$$\begin{aligned}
F_3^*(\lambda, \xi; \omega) = & \\
= & (\lambda^2 - \frac{1}{2}\omega^2) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot (1 - \xi) \right) + \\
& + \lambda^2 \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot (1 - \xi) \right) + \\
& + \omega^2 \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot \\
& \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right) + \\
& + \frac{\lambda^2 \cdot (\lambda^2 - \frac{1}{2}\omega^2)}{\omega^2 \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2}} \cdot \\
& \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right) - \\
& - \lambda^2 \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right) - \\
& - (\lambda^2 - \frac{1}{2}\omega^2) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \\
& \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right)
\end{aligned}$$

$$\Phi_4^{(0)}(\lambda, \xi; \omega) = \frac{F_4^*(\lambda, \xi; \omega)}{\det \Psi_1^*}, \quad (\lambda < \omega)$$

$$F_4^*(\lambda, \xi; \omega) =$$

$$\begin{aligned}
&= 2\lambda^2(\lambda^2 - \frac{1}{2}\omega^2) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot (1 - \xi) \right) + \\
&+ \lambda^2 \cdot (2\lambda^2 + \frac{3-\kappa}{1+\kappa}\omega^2) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot (1 - \xi) \right) + \\
&+ 2\lambda^2\omega^2 \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot \\
&\cdot \sin \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right) + \\
&+ \frac{\lambda^2 \cdot (\lambda^2 - \frac{1}{2}\omega^2) \cdot (2\lambda^2 + \frac{3-\kappa}{1+\kappa}\omega^2)}{\omega^2 \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2}} \cdot \\
&\cdot \sin \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \sin \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right) - \\
&- 2\lambda^4 \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \right) - \\
&- (\lambda^2 - \frac{1}{2}\omega^2) \cdot (2\lambda^2 + \frac{3-\kappa}{1+\kappa}\omega^2) \cdot \\
&\cdot \cos \left(\frac{\omega}{\alpha} \sqrt{\frac{\kappa-1}{\kappa+1}} \sqrt{1 - \frac{\kappa+1}{\kappa-1} \left(\frac{\lambda}{\omega}\right)^2} \cdot \xi \right) \cdot \cos \left(\frac{\omega}{\alpha} \sqrt{1 - \left(\frac{\lambda}{\omega}\right)^2} \right)
\end{aligned}$$

$$\Phi_5^{(0)}(\lambda, \xi; \omega) = \frac{F_3(\lambda, \xi; \omega)}{\det \Psi_1}, \quad (\lambda > \omega)$$

$$\begin{aligned}
F_3(\lambda, \xi; \omega) = & \\
= & (\lambda^2 - \frac{1}{2}\omega^2) \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \cdot (1 - \xi) \right) + \\
& + \lambda^2 \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \cdot (1 - \xi) \right) + \\
& + \sqrt{\lambda^2 - \omega^2} \cdot \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \cdot \\
& \cdot \sinh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \cdot \xi \right) \cdot \sinh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \right) + \\
& + \frac{\lambda^2 \cdot (\lambda^2 - \frac{1}{2}\omega^2)}{\sqrt{\lambda^2 - \omega^2} \cdot \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2}} \cdot \\
& \cdot \sinh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \cdot \xi \right) \cdot \sinh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \right) - \\
& - \lambda^2 \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \cdot \xi \right) \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \right) - \\
& - (\lambda^2 - \frac{1}{2}\omega^2) \cdot \\
& \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1} \omega^2} \cdot \xi \right) \cdot \cosh \left(\frac{1}{\alpha} \sqrt{\lambda^2 - \omega^2} \right)
\end{aligned}$$

$$\Phi_6^{(0)}(\lambda, \xi; \omega) = \frac{F_4(\lambda, \xi; \omega)}{\det \Psi_1}, \quad (\lambda > \omega)$$

$$\begin{aligned}
F_4(\lambda, \xi; \omega) = & \\
& = 2\lambda^2(\lambda^2 - \frac{1}{2}\omega^2) \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2} \cdot (1 - \xi)\right) + \\
& + \lambda^2 \cdot (2\lambda^2 + \frac{3 - \kappa}{\kappa + 1}\omega^2) \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2} \cdot (1 - \xi)\right) + \\
& + 2\lambda^2\sqrt{\lambda^2 - \omega^2} \cdot \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2} \cdot \\
& \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2} \cdot \xi\right) \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2}\right) + \\
& + \frac{\lambda^2 \cdot (\lambda^2 - \frac{1}{2}\omega^2) \cdot (2\lambda^2 + \frac{3 - \kappa}{\kappa + 1}\omega^2)}{\sqrt{\lambda^2 - \omega^2} \cdot \sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2}} \cdot \\
& \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2} \cdot \xi\right) \cdot \sinh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2}\right) - \\
& - 2\lambda^4 \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2} \cdot \xi\right) \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2}\right) - \\
& - (\lambda^2 - \frac{1}{2}\omega^2) \cdot (2\lambda^2 + \frac{3 - \kappa}{\kappa + 1}\omega^2) \cdot \\
& \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \frac{\kappa - 1}{\kappa + 1}\omega^2} \cdot \xi\right) \cdot \cosh\left(\frac{1}{\alpha}\sqrt{\lambda^2 - \omega^2}\right)
\end{aligned}$$

Функції для обчислення напруження на нижній грані

(у тому числі і det для різних Psi)

```
def Phi0_1(lamb, params):
```

```
    omega, _, _, alpha, kappa, _ = params
```

```
    f_s1 = F_s1(lamb, omega, alpha, kappa)
```

```
    psi_s1 = Psi_s1(lamb, omega, alpha, kappa)
```

```
    return f_s1 / psi_s1
```

```
def F_s1(lamb, omega, alpha, kappa):
```

```
    term1 = ((0.5 * omega**2 - lamb**2)
```

```
             * cmath.cos((omega / alpha)
```

```
             * cmath.sqrt(1 - (lamb / omega)**2)))
```

```

term2 = (lamb**2 * cmath.cos((omega / alpha)
    * cmath.sqrt((kappa - 1) / (kappa + 1))
    * cmath.sqrt(1 - ((kappa + 1) / (kappa - 1))
    * (lamb / omega)**2)))
return term1 + term2

```

```

def Psi_s1(lamb, omega, alpha, kappa):
    term1 = 2 * lamb**2 * (2 * lamb**2 - omega**2)
    term2 = ((4 * lamb**4 - 2 * lamb**2 * omega**2
        + 0.5 * omega**4)
        * cmath.cos((omega / alpha)
        * cmath.sqrt(1 - (lamb / omega)**2)))
    term2_1 = cmath.cos((omega / alpha)
        * cmath.sqrt((kappa - 1) / (kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1) / (kappa - 1))
        * (lamb / omega)**2))
    term3 = (lamb**2 * (4 * lamb**4
        - (2 * (3 * kappa + 1) / (kappa + 1))
        * lamb**2 * omega**2
        + (5 * kappa - 3) / (2 * (kappa + 1)) * omega**4))
    term3_1 = (omega**2 * cmath.sqrt((kappa - 1) / (kappa + 1))
        * cmath.sqrt(1 - (lamb / omega)**2)
        * cmath.sqrt(1 - ((kappa + 1) / (kappa - 1))
        * (lamb / omega)**2))
    term4 = cmath.sin((omega / alpha)
        * cmath.sqrt(1 - (lamb / omega)**2))
    term4_1 = cmath.sin((omega / alpha)
        * cmath.sqrt((kappa - 1) / (kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1) / (kappa - 1))
        * (lamb / omega)**2))
    return (term1 - term2 * term2_1
        + (term3 / term3_1) * term4 * term4_1)

```

```

def Phi0_2(lamb, params):
    omega, _, _, alpha, kappa, _ = params
    f_1 = F_1(lamb, omega, alpha, kappa)
    psi_1 = Psi_1(lamb, omega, alpha, kappa)
    return f_1 / psi_1

def F_1(lamb, omega, alpha, kappa):
    term1 = ((0.5 * omega**2 - lamb**2)
              * cmath.cosh((1 / alpha)
                           * cmath.sqrt(lamb**2 - omega**2)))
    term2 = lamb**2 * cmath.cosh((1 / alpha)
                                  * cmath.sqrt(lamb**2
                                                  - ((kappa - 1) / (kappa + 1)) * omega**2))
    return term1 + term2

def Psi_1(lamb, omega, alpha, kappa):
    term1 = 2 * lamb**2 * (2 * lamb**2 - omega**2)
    term2 = ((4 * lamb**4 - 2 * lamb**2 * omega**2
              + 0.5 * omega**4)
              * cmath.cosh((1 / alpha)
                           * cmath.sqrt(lamb**2 - omega**2))
              * cmath.cosh((1 / alpha) * cmath.sqrt(lamb**2
                                                      - ((kappa-1)/(kappa+1)) * omega**2)))
    term3 = (lamb**2 * (4 * lamb**4
                      - (2 * (3 * kappa + 1) / (kappa + 1))
                        * lamb**2 * omega**2
                      + (5 * kappa - 3) / (2 * (kappa + 1)) * omega**4))
    term4 = (cmath.sqrt(lamb**2 - omega**2)
              * cmath.sqrt(lamb**2 - ((kappa - 1) / (kappa + 1))
                            * omega**2))
    term5 = cmath.sinh((1 / alpha)
                       * cmath.sqrt(lamb**2 - omega**2))
    term6 = cmath.sinh((1 / alpha)
                       * cmath.sqrt(lamb**2 - ((kappa - 1) / (kappa + 1))
                                     * omega**2))

```

```

        * omega**2))
    return term1 - term2 + (term3 / term4) * term5 * term6

# Функції для обчислення напруження на циліндричній поверхні
def Phi0_3(lamb, xi, params):
    omega, _, _, alpha, kappa, rho = params
    return (F_s3(lamb, xi, omega, alpha, kappa)
            / Psi_s1(lamb, omega, alpha, kappa))

def F_s3(lamb, xi, omega, alpha, kappa):
    term1 = ((lamb**2 - 0.5 * omega**2) * cmath.cos((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2) * (1 - xi)))
    term2 = (lamb**2 * cmath.cos((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2) * (1 - xi)))
    term3 = (omega**2 * cmath.sqrt(1 - (lamb/omega)**2)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)
        * cmath.sin((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2) * xi)
        * cmath.sin((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)))
    term4 = ((lamb**2 * (lamb**2 - 0.5 * omega**2))
        / (omega**2 * cmath.sqrt(1 - (lamb/omega)**2)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)
        * cmath.sin((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))

```

```

        * (lamb/omega)**2) * xi)
        * cmath.sin((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2)))
term5 = (lamb**2 * cmath.cos((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2) * xi)
        * cmath.cos((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)))
term6 = ((lamb**2 - 0.5 * omega**2)
        * cmath.cos((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2) * xi)
        * cmath.cos((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2)))
result = term1 + term2 + term3 + term4 - term5 - term6
return result

```

```

def Phi0_4(lamb, xi, params):
    omega, _, _, alpha, kappa, rho = params
    return (F_s4(lamb, xi, omega, alpha, kappa)
            / Psi_s1(lamb, omega, alpha, kappa))

def F_s4(lamb, xi, omega, alpha, kappa):
    term1 = (2 * lamb**2 * (lamb**2 - 0.5 * omega**2)
            * cmath.cos((omega/alpha)
            * cmath.sqrt(1 - (lamb/omega)**2) * (1 - xi)))
    term2 = (lamb**2 * (2 * lamb**2 + (3 - kappa)/(1 + kappa)
            * omega**2) * cmath.cos((omega/alpha)
            * cmath.sqrt((kappa - 1)/(kappa + 1))
            * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
            * (lamb/omega)**2) * (1 - xi)))

```

```

term3 = (2 * lamb**2 * omega**2
        * cmath.sqrt(1 - (lamb/omega)**2)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)
        * cmath.sin((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2) * xi)
        * cmath.sin((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)))

term4 = ((lamb**2 * (lamb**2 - 0.5 * omega**2)
        * (2 * lamb**2 + (3 - kappa)/(1 + kappa) * omega**2))
        / (omega**2 * cmath.sqrt(1 - (lamb/omega)**2)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2))
        * cmath.sin((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2))
        * cmath.sin((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2) * xi))

term5 = (2 * lamb**4 * cmath.cos((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2) * xi)
        * cmath.cos((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))
        * (lamb/omega)**2)))

term6 = ((lamb**2 - 0.5 * omega**2)
        * (2 * lamb**2 + (3 - kappa)/(1 + kappa) * omega**2)
        * cmath.cos((omega/alpha)
        * cmath.sqrt((kappa - 1)/(kappa + 1))
        * cmath.sqrt(1 - ((kappa + 1)/(kappa - 1))

```

```

        * (lamb/omega)**2) * xi)
        * cmath.cos((omega/alpha)
        * cmath.sqrt(1 - (lamb/omega)**2)))
result = term1 + term2 + term3 + term4 - term5 - term6
return result

```

```

def Phi0_5(lamb, xi, params):
    omega, _, _, alpha, kappa, rho = params
    return (F_3(lamb, xi, omega, alpha, kappa)
            / Psi_1(lamb, omega, alpha, kappa))

def F_3(lamb, xi, omega, alpha, kappa):
    term1 = ((lamb**2 - 0.5 * omega**2)
              * cmath.cosh((1/alpha)
              * cmath.sqrt(lamb**2 - omega**2) * (1 - xi))
    term2 = (lamb**2 * cmath.cosh((1/alpha)
              * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
              * omega**2) * (1 - xi))))
    term3 = (cmath.sqrt(lamb**2 - omega**2)
              * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
              * omega**2)
              * cmath.sinh((1/alpha)
              * cmath.sqrt(lamb**2 - omega**2) * xi)
              * cmath.sinh((1/alpha)
              * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
              * omega**2)))
    term4 = ((lamb**2 * (lamb**2 - 0.5 * omega**2))
              / (cmath.sqrt(lamb**2 - omega**2)
              * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
              * omega**2))
              * cmath.sinh((1/alpha)
              * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
              * omega**2) * xi)

```

```

        * cmath.sinh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2)))
term5 = (lamb**2 * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2) * xi)
        * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2)))
term6 = ((lamb**2 - 0.5 * omega**2) * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2) * xi) * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2)))
result = term1 + term2 + term3 + term4 - term5 - term6
return result

```

```

def Phi0_6(lamb, xi, params):
    omega, _, _, alpha, kappa, rho = params
    return (F_4(lamb, xi, omega, alpha, kappa)
    / Psi_1(lamb, omega, alpha, kappa))

```

```

def F_4(lamb, xi, omega, alpha, kappa):
    term1 = (2 * lamb**2 * (lamb**2 - 0.5 * omega**2)
        * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2) * (1 - xi)))
    term2 = (lamb**2 * (2 * lamb**2 + (3 - kappa)/(kappa + 1)
        * omega**2) * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2) * (1 - xi)))
    term3 = (2 * lamb**2 * cmath.sqrt(lamb**2 - omega**2)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2) * cmath.sinh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2) * xi)
        * cmath.sinh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)

```

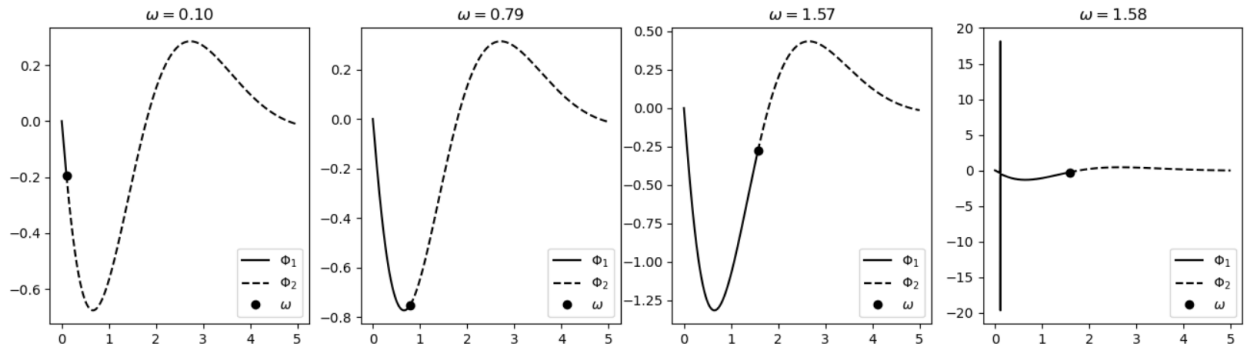
```

        * omega**2)))
term4 = ((lamb**2 * (lamb**2 - 0.5 * omega**2)
        * (2 * lamb**2 + (3 - kappa)/(kappa + 1)
        * omega**2))/(cmath.sqrt(lamb**2 - omega**2)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2)) * cmath.sinh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2) * xi) * cmath.sinh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2)))
term5 = (2 * lamb**4 * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2) * xi)
        * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2)))
term6 = ((lamb**2 - 0.5 * omega**2)
        * (2 * lamb**2 + (3 - kappa)/(kappa + 1) * omega**2)
        * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - (kappa - 1)/(kappa + 1)
        * omega**2) * xi)
        * cmath.cosh((1/alpha)
        * cmath.sqrt(lamb**2 - omega**2)))
result = term1 + term2 + term3 + term4 - term5 - term6
return result

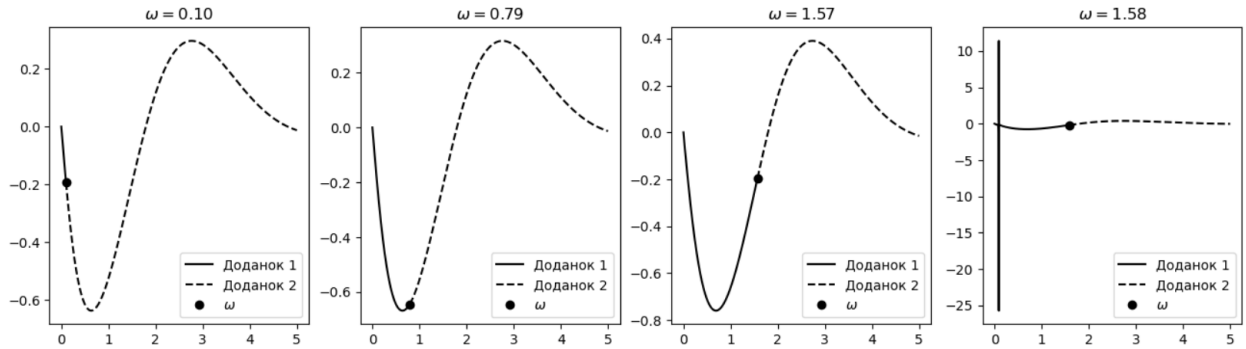
```

Додаток В. Графіки підінтегральних функцій

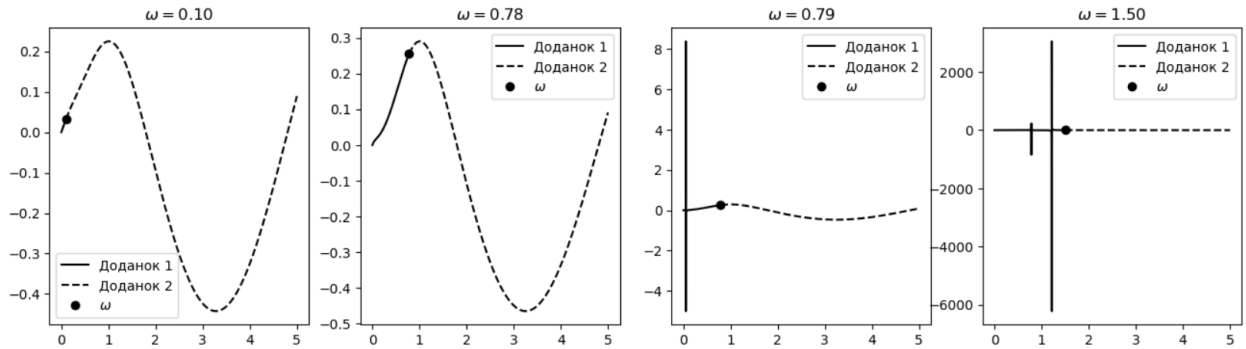
Графік підінтегральних функцій σ_ξ з $\mu = 0.15; \alpha = 1.0$



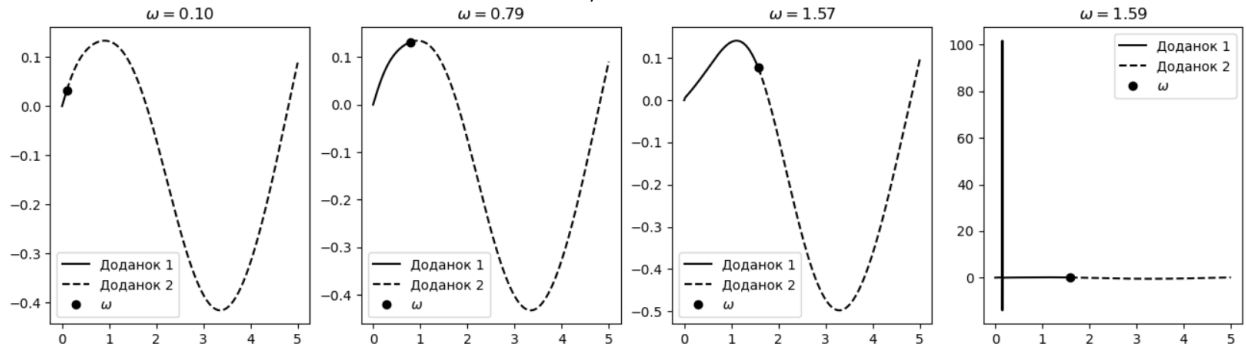
Графік підінтегральних функцій σ_ξ з $\mu = 0.40; \alpha = 1.0$



Графік підінтегральних функцій σ_ρ з $\mu = 0.25; \alpha = 0.5$



Графік підінтегральних функцій σ_ρ з $\mu = 0.25; \alpha = 1.0$



Графік функції σ_ρ з різними $\mu = 0.25$; та $\alpha = 0.5$

