

Зміст

Постановка задачі	4
Теоретичні відомості	5
Основна частина	16
1. Вступ	16
2. Відображення напівплощини $\text{Im}(\zeta) > 0$ на напівплощину $\text{Im}(z) > 0$ з відкинутим відрізком.....	17
3. Відображення напівплощини $\text{Im}(\zeta) > 0$ на напівплощину $\text{Im}(z) > 0$ зі сходинкою.....	19
4. Відображення напівплощини $\text{Im}(\zeta) > 0$ на область $\text{Im}(z) > 0 \cup (\text{Im}(z) < 0 \cap$ $\text{Re}(z) > 0)$ з відкинутим відрізком	21
5. Відображення напівплощини $\text{Im}(\zeta) > 0$ на напівплощину $\text{Im}(z) > 0$ з підйомом та відкинутим відрізком	23
6. Відображення смуги $-\pi < \text{Im}(\zeta) < \pi$ на площину з двома викинутими променями.....	25
7. Відображення смуги $-\pi < \text{Im}(\zeta) < \pi$ на дифузор	27
8. Відображення напівплощини $\text{Im}(\zeta) > 0$ на напівплощину $\text{Im}(z) > 0$ з відкинутим рівнобедреним трикутником.....	29
9. Чисельний метод знаходження функції струму в задачі обтікання тіла безвихровим потоком ідеальної нестисливої рідини	31
10. Опис програми	33
11. Обговорення отриманих результатів	49
Висновки	53
Список використаних джерел та літератури	55
Додатки	56
1. Лістинг кодів Wolfram Mathematica та Maple	56
2. Лістинг вихідного коду програми.....	58

Постановка задачі

1. Дослідити різні конформні відображення для розв'язування задач обтікання узагальнених багатокутників, за допомогою використання інтегралу Шварца-Кристоффеля.
2. Отримати формули для функцій конформного відображення таких областей, як напівплощина та смуга, комплексний потенціал для яких являє собою лінійну функцію, на узагальнені багатокутники.
3. Дослідити загальний метод знаходження функції струму в задачі обтікання тіла у вигляді багатокутника безвихровим потоком ідеальної нестисливої рідини, що базується на чисельному розв'язку задачі Діріхле для рівняння Лапласа.
4. Розробити програму для ілюстрації та аналізу отриманих результатів.

Теоретичні відомості

1. Плоска безвихрова течія ідеальної нестисливої рідини. Використання функцій комплексної змінної

При плоскому русі всі частинки рідини отримують переміщення, паралельне деякій площині, яку приймемо за площину xOy , причому у всіх паралельних площинах рух тотожний. Контур тіла, що обтікається, представимо як деяку лінію у площині, хоча насправді відбувається обтікання нескінченного циліндричного тіла. Оскільки рух безвихровий, то:

$$\vec{V} = \operatorname{grad} \varphi. \quad (1.1)$$

У разі нестисливої рідини рівняння збереження маси буде мати вигляд:

$$\operatorname{div} \vec{V} = 0. \quad (1.2)$$

З урахуванням (1.1) та (1.2) маємо рівняння Лапласа для функції φ , яке в плоскому випадку буде мати вигляд:

$$\Delta^2 \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0. \quad (1.3)$$

В задачі обтікання тіла плоским, однорідним на нескінченності потоком зі швидкістю \vec{V}_∞ граничні умови будуть складатися з умов непротікання границь тіла:

$$V_n = \frac{\partial \varphi}{\partial n} = 0 \text{ на контурі тіла } C \quad (1.4)$$

та умов на нескінченності

$$u = \frac{\partial \varphi}{\partial x} = V_\infty * \cos \theta_\infty, v = \frac{\partial \varphi}{\partial y} = V_\infty * \sin \theta_\infty,$$

де θ_∞ – кут між вектором швидкості \vec{V}_∞ та віссю Ox .

Задача такого роду відноситься до класичних задач Неймана, розв'язанням якої посвячено багато математичних досліджень.

З рівняння нерозривності

$$\operatorname{div} \vec{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (1.5)$$

вітікає те, що завжди можна знайти функцію $\psi(x, y)$, що тотожно задовольняє рівнянню (1.5) та яка пов'язана з проекціями швидкості u і v рівностями:

$$u = \frac{\partial \psi}{\partial y}, v = -\frac{\partial \psi}{\partial x}. \quad (1.6)$$

Функція $\psi(x, y)$ має простий гідродинамічний сенс. Дійсно, запишемо диференційне рівняння ліній струму для плоского стаціонарного руху:

$$\frac{dx}{u} = \frac{dy}{v},$$

і підставимо в нього значення проекцій швидкості по (1.6). Маємо:

$$\frac{dx}{\partial \psi / \partial y} = \frac{dy}{-\partial \psi / \partial x}$$

або

$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial y} dy = 0.$$

З останньої рівності витікає, що функція ψ зберігає постійне значення впродовж ліній струму. Інакше кажучи, сімейство ліній рівня функції

$$\psi(x, y) = C \quad (1.7)$$

є сукупністю ліній струму. Функція $\psi(x, y)$ називається функцією струму, а функція $\varphi(x, y)$ – потенціалом.

Порівняємо вирази проекцій швидкості через потенціал швидкостей, які у випадку плоского руху мають вигляд:

$$u = \frac{\partial \varphi}{\partial x}, v = \frac{\partial \varphi}{\partial y}, \quad (1.8)$$

та вирази (1.6) тих же проекцій через функцію струму ψ ; будемо мати наступну систему співвідношень:

$$\frac{\partial \varphi}{\partial x} = \frac{\partial \psi}{\partial y}, \frac{\partial \varphi}{\partial y} = -\frac{\partial \psi}{\partial x}. \quad (1.9)$$

Ці рівняння виражають умови Коші-Рімана, при виконанні яких комплексна величина

$$W = \varphi(x, y) + i\psi(x, y) \quad (1.10)$$

Буде аналітичною функцією однієї комплексної змінної $z = x + iy$.

Функцію $W(z)$, що об'єднує відповідно до (1.10) в один комплекс два потенціали: скалярний – потенціал швидкостей – і проекцію векторного – функцію струму, називають комплексним потенціалом течії.

Покажемо, як, знаючи комплексний потенціал $W(z)$, знайти вектор швидкості \vec{V} або його проекції u та v . Як відомо, кожному комплексному числу можна поставити у відповідність вектор на площині з проекціями, що дорівнюють дійсній та уявним частинам цього комплексного числа. Будемо літерою V позначувати комплексну швидкість

$$V = u + iv, \quad (1.11)$$

а для величини швидкості збережемо звичайне позначення модуля комплексного числа:

$$|V| = \sqrt{u^2 + v^2}. \quad (1.12)$$

Також введемо в розгляд спряжену швидкість \bar{V} , що дорівнює

$$\bar{V} = u - iv. \quad (1.13)$$

Якщо θ – кут, утворений вектором комплексної швидкості V з дійсною віссю, то

$$\begin{cases} V = u + iv = |V|(cos\theta + isin\theta) = |V|e^{i\theta}, \\ \bar{V} = u - iv = |V|(cos\theta - isin\theta) = |V|e^{-i\theta}. \end{cases} \quad (1.14)$$

Спряжені швидкості мають ту ж величину (модуль), що й комплексна швидкість, але направлена по дзеркальному відображення комплексної швидкості відносно дійсної вісі (рис. 1).

Розглянемо похідну dW/dz комплексного потенціала по комплексному аргументу:

$$\frac{dW}{dz} = \frac{dW}{dx} = \frac{\partial(\varphi+i\psi)}{\partial x} = \frac{\partial\varphi}{\partial x} + i \frac{\partial\psi}{\partial x},$$

звідси, відповідно до (1.6) та (1.8), відразу слідує, що

$$\frac{dW}{dz} = u - iv = \bar{V} = |V|e^{-i\theta}, \quad (1.15)$$

тобто похідна від комплексного потенціалу по комплексній координаті дорівнює спряженій швидкості.

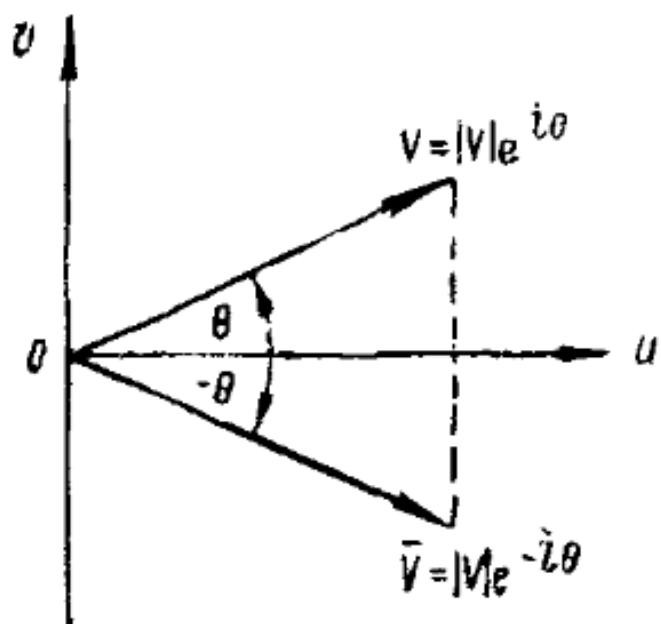


Рис. 1.

2. Комплексний потенціал однорідного потоку

Розглянемо лінійну функцію $W(z)=az$, де a – в загальному випадку комплексна константа. Будуючи спряжену швидкість:

$$\bar{V} = \frac{dW}{dz} = a = \text{const} = u_{\infty} - iv_{\infty} = |V_{\infty}|(\cos\alpha - i\sin\alpha) = |V_{\infty}|e^{-i\alpha},$$

бачимо, що комплексна константа являє собою однакову за модулем та напрямком у всьому потоці спряжену швидкість. Однаковою буде й комплексна швидкість:

$$V = V_{\infty} = u_{\infty} + iv_{\infty} = |V_{\infty}|(\cos\alpha + i\sin\alpha) = |V_{\infty}|e^{i\alpha}.$$

Таким чином, лінійна функція визначає комплексний потенціал однорідного потоку зі швидкістю V_{∞} , що нахилений до дійсної осі фізичної площини під кутом α (рис. 2).

$$W = (u_{\infty} - iv_{\infty})z = |V_{\infty}|ze^{-i\alpha} = |V_{\infty}|z(\cos\alpha - i\sin\alpha). \quad (2.1)$$

Відокремлюючи дійсну та уявну частини, знайдемо потенціал швидкостей

$$\varphi = |V_{\infty}|(x\cos\alpha + y\sin\alpha) = u_{\infty}x + v_{\infty}y,$$

та функцію струму

$$\psi = |V_{\infty}|(-x\sin\alpha + y\cos\alpha) = -u_{\infty}x + v_{\infty}y.$$

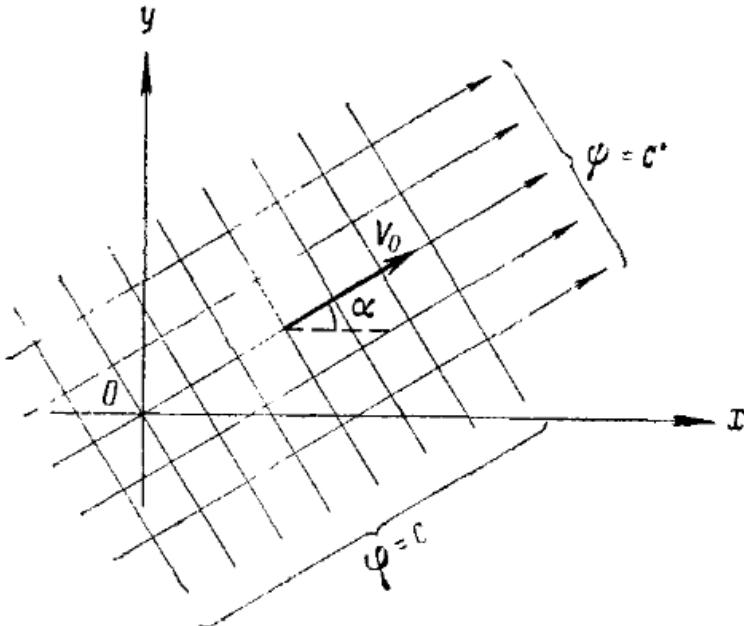


Рис. 2

3. Плоска задача про рух тіла в ідеальній рідині

Знаходження комплексного потенціалу

$$w(z) = \varphi(x, y) + i\psi(x, y), \quad (3.1)$$

що визначає плоский безвихровий рух ідеальної нестисливої рідини, може бути зведено до знаходження однієї функції струму $\psi(x, y)$, так як потенціал $\varphi(x, y)$ пов'язаний з функцією струму умовами Коші-Рімана. Функція струму $\psi(x, y)$, яка у всіх точках потоку нестисливої рідини вважається неперервною, задовольняє в цих точках рівнянню Лапласа $\Delta\psi = 0$, а на границях потоку – деяким відомим умовам.

Якщо розглядається плоский рух циліндричного тіла в нерухомій рідині, що займає нескінченну область, то граничними умовами для функції струму будуть:

$$\frac{\partial\psi}{\partial x} = 0, \frac{\partial\psi}{\partial y} = 0 \quad (3.2)$$

для нескінченно далеких точок потоку, так як швидкість в них повинна дорівнювати нулю. В кожній точці контуру тіла повинні співпадати нормальні проекції швидкостей u_n – власне контуру та v_n – частинки рідини, що прилягає до тіла. Помітивши, що

$$v_n = v_x \sin\theta - v_y \cos\theta = v_x \frac{dy}{ds} - v_y \frac{dx}{ds} = \frac{\partial\psi}{\partial y} \frac{dy}{ds} + \frac{\partial\psi}{\partial x} \frac{dx}{ds} = \frac{d\psi}{ds}, \quad (3.3)$$

де θ є кут між елементом ds лінії струму та віссю Ox , ми можемо цю умову виразити співвідношенням:

$$\frac{d\psi}{ds} = u_n = u_x \sin\theta - u_y \cos\theta. \quad (3.4)$$

Якщо при цьому в рідині є також нерухомі тіла, то очевидно, що на їх контурах нормальні складові швидкості прилягаючих до тіла частинок повинна дорівнювати нулю або, інакше кажучи, сам нерухомий контур повинен бути в зіткненні з однією або декількома лініями струму. В цьому випадку додаємо ще одну граничну умову:

$$\frac{d\psi}{ds} = 0, \psi = const, \quad (3.5)$$

для точок нерухомого контуру.

Якщо набігаючий потік, швидкість якого в нескінченно віддалених точках дорівнює U і направлена вздовж вісі Ox , обтікає нерухоме тіло, то граничними умовами будуть:

$$\psi = Uy + c \quad (3.6)$$

для нескінченно віддалених точок та

$$\frac{d\psi}{ds} = 0, \psi = const \quad (3.7)$$

для точок контуру.

Задача про визначення в деякій області D функції $\psi(x, y)$, що задовольняє рівнянню Лапласа, по відомим значенням функції на контурі області, називається задачею Діріхле, тобто визначення плоского безвихрового руху нестисливої рідини зводиться до розв'язання деякої задачі Діріхле.

4. Поняття конформного відображення. Принцип симетрії

Нехай задано неперервне та взаємно-однозначне відображення області D в площині z на деяку область D^* в площині w :

$$w = f(z) = u(x, y) + iv(x, y). \quad (4.1)$$

Припустимо, що функції $u(x, y)$ та $v(x, y)$ диференційовані в цій області. Фіксуємо довільну точку z_0 із D і в околі цієї точки замінимо приріст функцій $u(x, y)$ та $v(x, y)$ диференціалами. За означенням диференціала приріст можна представити у вигляді

$$\begin{cases} u - u_0 = \frac{\partial u}{\partial x}(x - x_0) + \frac{\partial u}{\partial y}(y - y_0) + \eta_1 \Delta r, \\ v - v_0 = \frac{\partial v}{\partial x}(x - x_0) + \frac{\partial v}{\partial y}(y - y_0) + \eta_2 \Delta r, \end{cases} \quad (4.2)$$

де частинні похідні беруться в точці z_0 , $\Delta r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$, а η_1, η_2 прямають до нуля при $\Delta r \rightarrow 0$. Заміна приростів диференціалами зводиться до відкидання в співвідношеннях (4.2) членів $\eta_1 \Delta r, \eta_2 \Delta r$, які є малими величинами більш високого порядку малості, ніж решта членів цих формул.

Геометрично ця заміна рівносильна заміні відображення $w = f(z)$ відображенням

$$\begin{cases} u - u_0 = \frac{\partial u}{\partial x}(x - x_0) + \frac{\partial u}{\partial y}(y - y_0), \\ v - v_0 = \frac{\partial v}{\partial x}(x - x_0) + \frac{\partial v}{\partial y}(y - y_0), \end{cases} \quad (4.3)$$

яке називається головною лінійною частиною відображення (3.1).

Відображення (4.3) можна переписати у вигляді:

$$\begin{cases} u = ax + by + l, \\ v = cx + dy + m, \end{cases} \quad (4.4)$$

де

$$\begin{cases} a = \frac{\partial u}{\partial x}, b = \frac{\partial u}{\partial y}, c = \frac{\partial v}{\partial x}, d = \frac{\partial v}{\partial y}, \\ l = u_0 - \frac{\partial u}{\partial x}x_0 - \frac{\partial u}{\partial y}y_0, m = v_0 - \frac{\partial v}{\partial x}x_0 - \frac{\partial v}{\partial y}y_0 \end{cases} \quad (4.5)$$

не залежать від x та y . Воно являє собою так зване лінійне перетворення площини (x, y) .

Взаємно-однозначне відображення (4.1) області D на область D^* називається конформним, якщо в околі будь-якої точки D головна лінійна частина цього відображення є ортогональне перетворення, що зберігає орієнтацію. З цього означення витікають дві основні властивості конформних відображень:

- 1) Конформне відображення перетворює нескінченно малі кола в кола з точністю до малих величин більшого порядку.
- 2) Конформне відображення зберігає кути між кривими в точках їх перетину.

Перша властивість означає, що при малих r коло $C: |z - z_0| = r$ переходить в криву C^* таку, що відстань будь-якої її точки від кола $|w - w_0| = \rho$, проведеної через будь-яку точку кривої C^* – образа C при відображені, що розглядається, є малою величиною більшого порядку відносно r .

Друга властивість означає, що кут в точці z_0 між будь-якими кривими Γ_1 та Γ_2 дорівнює куту в точці w_0 між образами Γ_1^* та Γ_2^* цих кривих. Для доведення цієї властивості достатньо помітити, що під кутом між кривими мається на увазі кут між їх дотичними і що головна лінійна частина диференційованого відображення переводить дотичну до кривої Γ_k у дотичну до кривої Γ_k^* .

Запишемо умови того, що відображення (4.1) буде конформним у вигляді:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad (4.6)$$

причому повинно бути:

$$\Delta = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 = |f'(z_0)|^2 \neq 0, \quad (4.7)$$

бо при $\Delta = 0$ головна лінійна частина відображення $w = f(z)$ є виродженою, що суперечить умові того, що відображення є конформним. Таким чином, умови того, що відображення є конформним, співпадають з умовами Коші-Рімана диференційованості (аналітичності) функції $f(z)$ в області D , причому нерівність (4.7) показує, що похідна $f'(z)$ повинна бути відмінною від нуля.

Далі маємо

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = \sqrt{\Delta} \cos \alpha, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = \sqrt{\Delta} \sin \alpha,$$

звідки легко отримати геометричну інтерпретацію похідної від функції комплексної змінної. Ми маємо:

$$|f'(z)| = \sqrt{\Delta}; \arg f'(z) = \alpha, \quad (4.8)$$

тобто модуль та аргумент похідної $f'(z)$ означають відповідно коефіцієнт розтягу і кут повороту головної лінійної частини відображення $w = f(z)$ в точці z або, інакше кажучи, коефіцієнт розтягу і кут повороту самого відображення $w = f(z)$ в точці z .

Теорема (Б. Ріман, 1851 р.). Які б не були однозв'язні області D та D^* (з границями, що складаються більш ніж з однієї точки) і як би не були задані точки z_0 із D та w_0 із D^* та дійсне число α_0 , існує одне й тільки одне конформне відображення

$$w = f(z)$$

області D на область D^* таке, що

$$f(z_0) = w_0, \arg f'(z_0) = \alpha_0.$$

Принцип симетрії дає в одному частинному випадку просту достатню умову існування аналітичного продовження функції, що реалізує конформне відображення.

Теорема (Б. Ріман, Г. Шварц). Нехай границя області D_1 містить дугу кола C і нехай функція $w = f_1(z)$ реалізує конформне відображення цієї області на область D_1^* таке, ще дуга C переходить в ділянку C^* , границі D_1^* , що також є дугою кола. В цих умовах функція $f_1(z)$ допускає аналітичне продовження $f_2(z)$ через дугу C в область D_2 , симетричну з D_1 відносно C , причому функція $w = f_2(z)$ реалізує конформне відображення області D_2 на область D_2^* , симетричну з D_1^* відносно C^* , а функція

$$w = f(z) = \begin{cases} f_1(z) & \text{в } D_1, \\ f_1(z) = f_2(z) & \text{на } C, \\ f_2(z) & \text{в } D_2 \end{cases}$$

реалізує конформне відображення області $D_1 + C + D_2$ на область $D_1^* + C^* + D_2^*$.

5. Інтеграл Шварца-Крістоффеля та теорема Рімана

Теорема 1 (Г. Шварц, Е. Крістоффель, 1867-1869 рр.). Якщо функція $w = f(z)$ реалізує конформне відображення верхньої півплощини $\operatorname{Im}(z) > 0$ на внутрішність обмеженого багатокутника Δ з кутами $a_k\pi$ ($0 < a_k \leq 2, k = 1, 2, \dots, n$) при вершинах, причому відомі точки a_k дійсної вісі ($-\infty < a_1 < a_2 < \dots < a_n < \infty$), що відповідають вершинам цього багатокутника, то $f(z)$ є інтегралом

$$f(z) = C \int_{z_0}^z (z - a_1)^{a_1-1} (z - a_2)^{a_2-1} \dots (z - a_n)^{a_n-1} dz + C_1,$$

де z_0, C, C_1 – деякі константи.

Теорема 2. Існує одне й тільки одне конформне відображення $w = f(z)$ області D на область D^* , що переводить три граничні точки z_k області D в три граничні точки w_k області D^* . Точки z_k і w_k задаються довільно, але з дотриманням порядку проходження при обході границь областей.

Інтеграл Шварца-Крістоффеля отримано в припущеннях, що точки a_k , що відповідають вершинам багатокутника, відомі. Але в задачах на конформні відображення задаються лише точки A_k багатокутника, а точки a_k лишаються невідомими. Відповідно до теореми 2 цього пункту, три з них можна задавати довільно, а інші точки та константи C і C_1 потрібно визначати з умов задачі.

Формула Шварца-Крістоффеля залишається в силі також для багатокутників, у яких одна або більше вершин лежать в нескінченно віддаленій точці, якщо при цьому кут між двома прямими з вершиною у нескінченості визначається як кут в кінечній точці їх перетину, що взятий зі знаком мінус.

Основна частина

Вступ

Метою даної роботи є дослідження методів візуалізації та розрахунку безвихрової течії ідеальної нестисливої рідини навколо узагальнених багатокутників. Незважаючи на те, що реальні процеси обтікання часто характерні наявністю турбулентності, а сама рідина має в'язкість, модель ідеальної рідини використовується в різних розділах гідродинаміки. Наприклад, при дослідженні в'язкого примежового шару на одній межі шару швидкість руху частинок приймається нульовою, що відповідає умові прилипання частинок рідини до поверхні тіла, що обтікається, а швидкість на другій межі можна прийняти рівною швидкості потенціального потоку, що знайдена в рамках моделі ідеальної рідини. Дано робота характерна тим, що в ній розглядається два різні підходи дослідження руху рідини, кожен з яких має свої переваги та недоліки. Новизна роботи полягає у створенні програмного забезпечення, яке може візуалізувати ці підходи та яке можна розширювати на більш широкий клас задач.

1. Відображення напівплощини $\operatorname{Im}(\zeta) > 0$ на напівплощину $\operatorname{Im}(z) > 0$ з відкинутим відрізком

Спочатку отримаємо зворотне відображення, тобто відображення напівплощини з відкинутим відрізком на напівплощину (рис. 1).

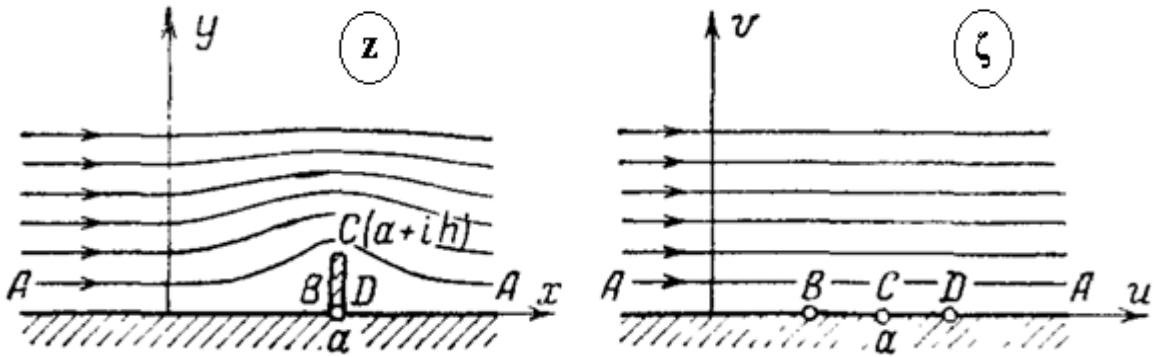


Рис. 1.

Нехай з півплощини $\operatorname{Im}(z) > 0$ виключений відрізок $(a, a + ih)$. Для отримання відображення ми скористаємося тим, що відображення $\zeta = z^2$ подвоює кути в початку координат та, отже, може «розправити» кут між виключеним відрізком та віссю x .

Відповідно до цього ми зрушуємо напівплощину z на відрізок a вліво: $z_1 = z - a$ та, застосовуючи відображення $z_2 = z_1^2$, отримуємо напівплощину з відкинутим променем $-h^2 < \operatorname{Re}(z_2) < \infty, \operatorname{Im}(z_2) = 0$. Потім ми знову зрушуємо площину z_2 на величину h^2 вправо: $z_3 = z_2 + h^2$. Застосовуючи відображення $z_4 = \sqrt{z_3}$, отримаємо верхню напівплощину. Таким чином, шукане відображення має вигляд:

$$z_4 = \sqrt{(z - a)^2 + h^2}.$$

Зрушуючи що площину z_4 на відрізок a вправо, щоб точка $z = a + ih$ перейшла в точку a , остаточно отримаємо:

$$\zeta = \sqrt{(z - a)^2 + h^2} + a. \quad (1.1)$$

Дуже легко можна отримати зворотне конформне відображення $z = z(\zeta)$, виразивши в (1.1) z через ζ :

$$z = a \pm \sqrt{a^2 - 2a\zeta - h^2 + \zeta^2}. \quad (1.2)$$

Незважаючи на те, що вираз (1.1) не вирішує вихідну задачу, він видається дуже важливим, коли потрібно розрахувати швидкості у фізичній площині z . Як відомо:

$$\bar{V} = \frac{dW}{dz} = \frac{dW}{d\zeta} \zeta'_z = \frac{dW}{d\zeta} \frac{1}{z'_\zeta},$$

де z'_ζ розраховується в точці $\zeta = \zeta(z)$, а оскільки, зазвичай, задається положення точки z , то функцію $\zeta = \zeta(z)$ потрібно знати. Зазначимо, що розглянута вище задача є єдиним прикладом конформного відображення в даній роботі, де функцію $\zeta = \zeta(z)$ можна було легко виразити. В подальших випадках для знаходження точки ζ_0 , такої, що $\zeta_0 = \zeta(z_0)$ було застосовано метод Ньютона розв'язання нелінійних рівнянь для рівняння $z(\zeta) - z_0 = 0$ відносно ζ .

Слід також зазначити, що знак перед радикалом в формулі (1.2) обирається в залежності від того, чи належить шукане значення z верхній напівплощині.

2. Відображення напівплощини $\operatorname{Im}(\zeta) > 0$ на напівплощину $\operatorname{Im}(z) > 0$ зі сходинкою

Відображення напівплощини на трикутник, що має вигляд сходинки (рис. 2, а, б), згідно з інтегралом Шварца-Кристоффеля, будемо шукати в наступному вигляді:

$$z = C \int_1^\zeta (\zeta + 1)^{\frac{1}{2}} (\zeta - 1)^{-\frac{1}{2}} d\zeta, \quad (2.1)$$

де постійна $C_1 = 0$ в силу того, що тоці $a_3 = 1$ відповідає точка $A_3 = 0$ на фізичній площині.

Проінтегруємо праву частину (2.1):

$$z = C \int_1^\zeta \frac{\sqrt{\zeta + 1}}{\sqrt{\zeta - 1}} d\zeta = C \int_1^\zeta \frac{\zeta d\zeta}{\sqrt{\zeta^2 - 1}} + C \int_1^\zeta \frac{d\zeta}{\sqrt{\zeta^2 - 1}} = C \left(\sqrt{\zeta^2 - 1} + \operatorname{Arch} \zeta \right).$$

Постійну C визначимо із відповідності токок a_2 і A_2 :

$$ih = C\pi i; C = \frac{h}{\pi}$$

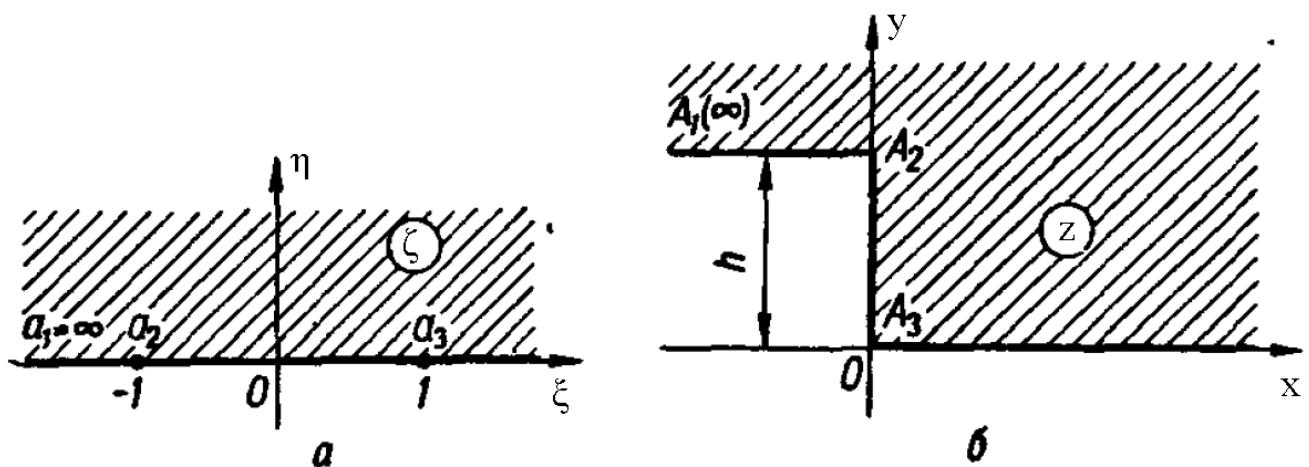


рис. 2

Отже, відображуюча функція має наступний вигляд:

$$z = \frac{h}{\pi} \left(\pm \sqrt{\zeta^2 - 1} + \operatorname{Arch} \zeta \right). \quad (2.2)$$

Якщо розглядати відображення напівплощини з підйомом (рис. 3),

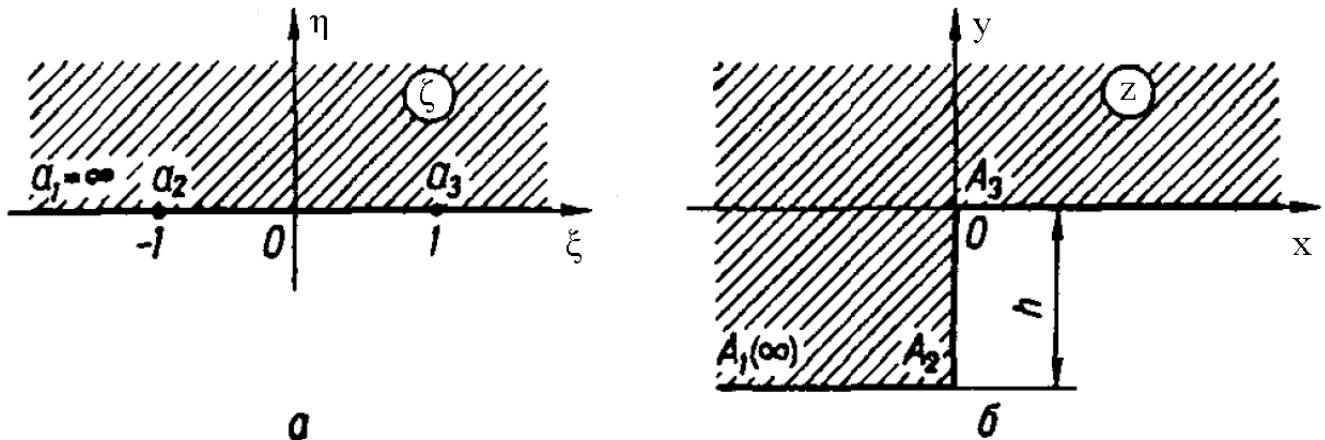


рис. 3

то з інтегралу Шварца-Кристоффеля маємо:

$$z = C \int_1^{\zeta} \frac{\sqrt{\zeta-1}}{\sqrt{\zeta+1}} d\zeta = C(\sqrt{\zeta^2-1} - \operatorname{Arch}\zeta)$$

або, визначивши постійну С із відповідності точок a_2 і A_2 , остаточно отримаємо:

$$z = \frac{h}{\pi} (\pm \sqrt{\zeta^2 - 1} - \operatorname{Arch}\zeta). \quad (2.3)$$

3. Відображення напівплощини $\operatorname{Im}(\zeta) > 0$ на область $\operatorname{Im}(z) > 0 \cup (\operatorname{Im}(z) < 0 \cap \operatorname{Re}(z) > 0)$ з відкинутим відрізком

Розглянемо відображення верхньої напівплощини на трикутник, зображеній на рис.4,б.

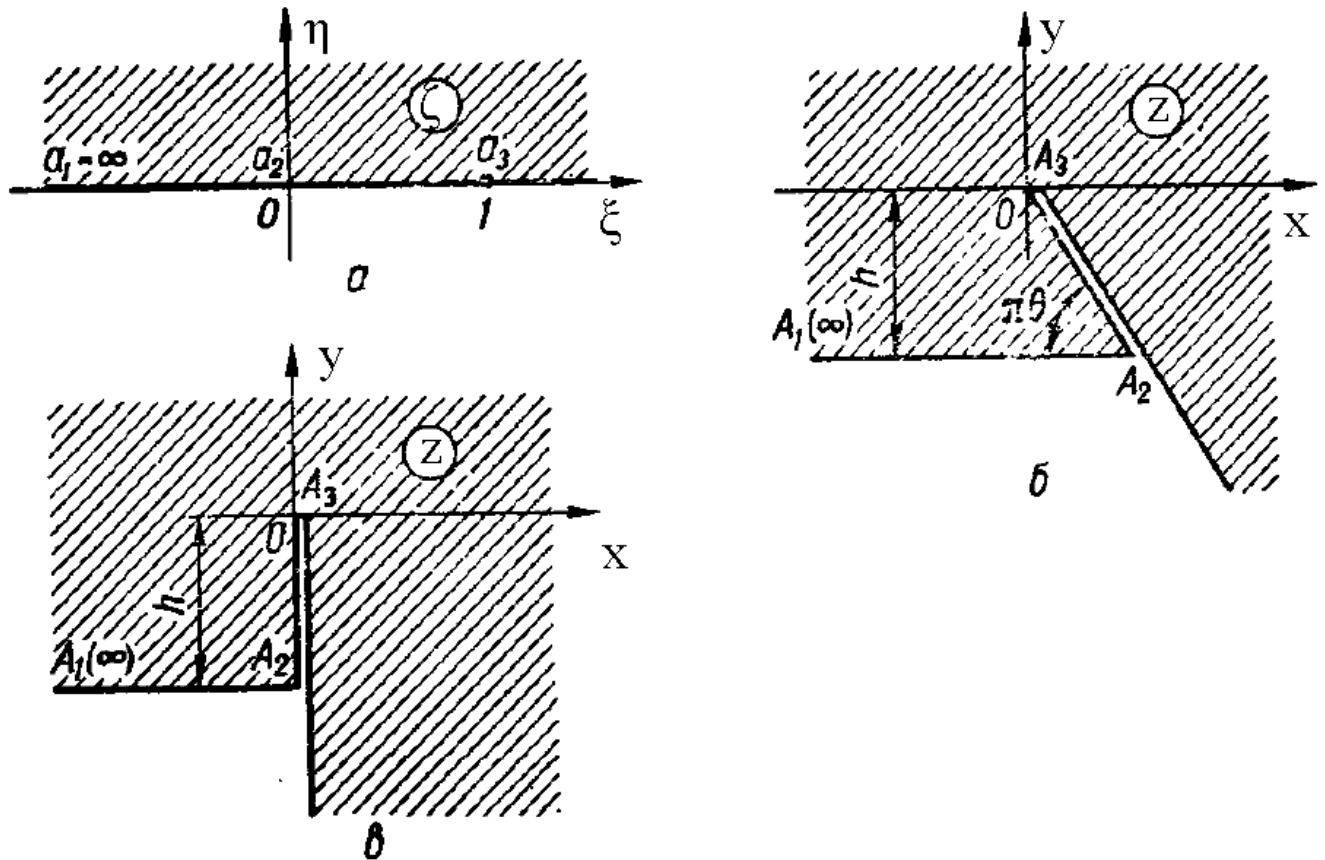


рис.4

У випадку, коли $\theta = \frac{1}{2}$ (рис.4,в), відображаюча функція приймає наступний вигляд:

$$z = C \int_0^{\zeta} \frac{\zeta - 1}{\sqrt{\zeta}} d\zeta + C_1 = \frac{2C}{3} \sqrt{\zeta} (\zeta - 3) + C_1. \quad (3.1)$$

Постійні С та C_1 визначимо з відповідності точок $a_2 = 0$ і $A_2 = h(\operatorname{ctg}\pi\theta - i)$, а також $a_3 = 1$ і $A_3 = 0$:

$$-ih = C_1;$$

$$0 = -\frac{4}{3}C + C_1,$$

звідки:

$$C_1 = -ih; C = -\frac{3}{4}ih.$$

Остаточно відображуюча функція запишеться у вигляді:

$$z = -ih \left(\sqrt{\zeta} \frac{\zeta-3}{2} + 1 \right). \quad (3.2)$$

4. Відображення напівплощини $\operatorname{Im}(\zeta) > 0$ на напівплощину $\operatorname{Im}(z) > 0$ з підйомом та відкинутим відрізком

Запишемо для відображення верхньої напівплощини на чотирикутник (рис.5) із заданими параметрами h_1 і h_2 інтеграл Шварца-Кристоффеля:

$$z = C \int_0^\zeta (\zeta + 1)^{-\frac{1}{2}} (\zeta - \lambda)(\zeta - 1)^{-\frac{1}{2}} d\zeta + C_1 = C \int_0^\zeta \frac{\zeta - \lambda}{\sqrt{\zeta^2 - 1}} d\zeta + C_1 \quad (4.1)$$

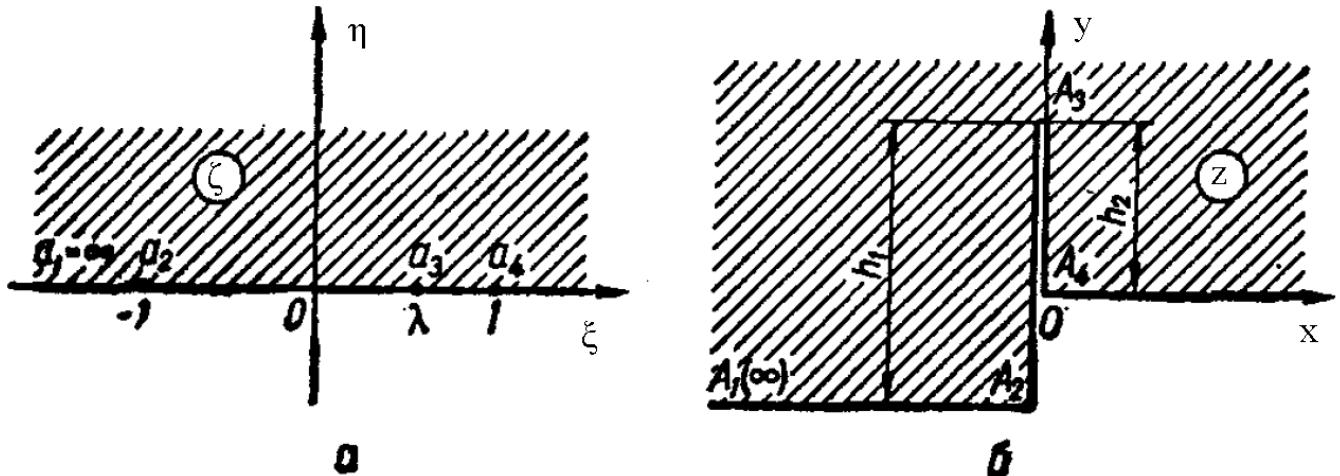


рис.5

або після інтегрування

$$z = C \left(\sqrt{\zeta^2 - 1} - \lambda \operatorname{Arch} \zeta \right) + C'_1. \quad (4.2)$$

Постійні C_1 і C'_1 знайдемо із відповідності точок $a_2 = -1$ і $A_2 = -i(h_1 - h_2)$, а також $a_4 = 1$ і $A_4 = 0$:

$$\begin{aligned} -i(h_1 - h_2) &= -C \lambda \operatorname{Arch}(-1) + C_1 = -iC \lambda \pi + C'_1; \\ 0 &= -C \lambda \operatorname{Arch} 1 + C'_1 = C'_1, \end{aligned}$$

Звідки

$$C'_1 = 0, C = \frac{h_1 - h_2}{\lambda \pi}. \quad (4.3)$$

Підставивши (4.3) в (4.2) знайдемо:

$$z = \frac{h_1 - h_2}{\lambda \pi} \left(\sqrt{\zeta^2 - 1} - \lambda \operatorname{Arch} \zeta \right). \quad (4.4)$$

Із відповідності точок $a_3 = \lambda$ і $A_3 = ih_2$ отримаємо рівняння для видначення константи λ :

$$\frac{h_2\pi}{h_1-h_2} = \frac{\sqrt{1-\lambda^2}}{\lambda} - \arccos\lambda \quad (-1 < \lambda < 1). \quad (4.5)$$

5. Відображення смуги $-\pi < \operatorname{Im}(\zeta) < \pi$ на площину з двома відкинутими променями

Результат такого відображення зображений на рис. 6. Застосуємо принцип симетрії – верхня половина області в площині z являє собою трикутник з двома вершинами у нескінченості та кутами $\alpha_1 = \alpha - 1, \alpha_2 = -\alpha, \alpha_3 = 2$. Щоб скористатися інтегралом Шварца-Кристоффеля, відобразимо полосу $0 < y < \pi$ на напівплощину $q = e^{\zeta}$.

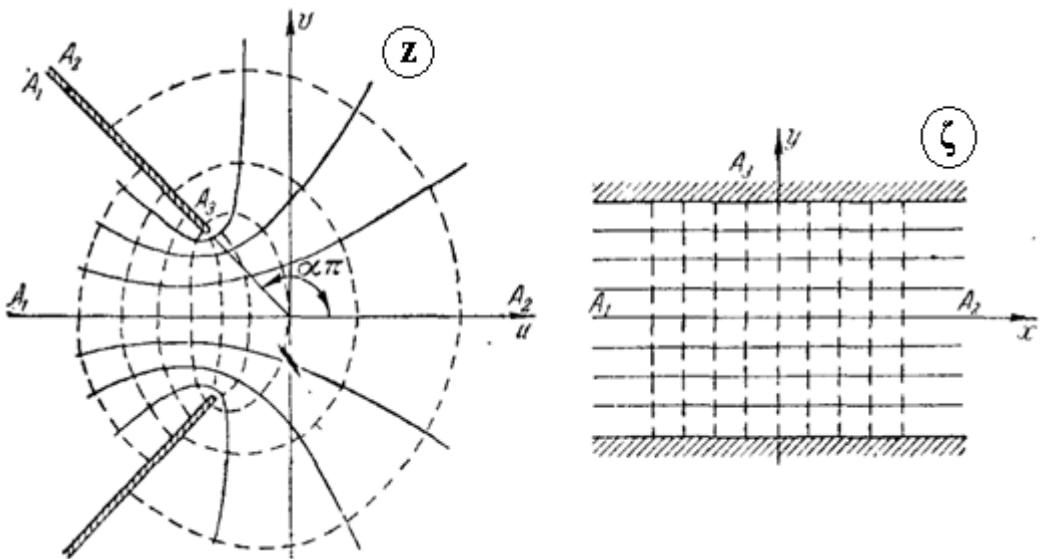


Рис. 6

Враховуючи відповідність точок, що вказана на рис. 6, ми приймаємо $a_1 = 0, a_2 = -\infty$, тоді точка a_3 попаде на від'ємну піввісь і ми вважаємо $a_3 = -a$, де a – поки що невідоме додатнє число. Інтеграл Шварца-Кристоффеля приймає вид:

$$z = C \int q^{\alpha-2} (q + a) dq + C_1 = C \left(\frac{q^\alpha}{\alpha} + \frac{a}{\alpha-1} q^{\alpha-1} \right) + C_1; \quad (5.1)$$

тут C – додатня константа, тому що промінь $A_1 A_2$ не повертається при відображені і, отже, $\arg C = 0$. C_1 – дійсне, так як підстановка в (5.1) додатніх значень q повинна привести до дійсних w . Щоб формула (5.1) прийняла найбільш простий вигляд, положимо $\frac{1}{\alpha} = -\frac{a}{\alpha-1}$, тобто $a = \frac{1-\alpha}{\alpha}$; будемо мати:

$$z = \frac{C}{\alpha} (q^\alpha - q^{\alpha-1}) + C_1. \quad (5.2)$$

Відповідність точок $q = -a$ та $w = le^{i\alpha\pi}$ дає:

$$\frac{C}{\alpha} (e^{i\alpha\pi} a^\alpha + e^{i\alpha\pi} a^{\alpha-1}) + C_1 = l e^{i\alpha\pi},$$

звідки, враховуючи, що константи C і C_1 – дійсні, отримаємо значення $C_1 = 0$ і $C = \frac{\alpha l}{a^{\alpha-1}(a+1)} = l\alpha^{\alpha+1}(1-\alpha)^{1-\alpha}$. Підставляючи в (5.2) $q = e^\zeta$, знаходимо шукане відображення:

$$z = l\alpha^\alpha(1-\alpha)^{1-\alpha}(e^{\alpha\zeta} - e^{(\alpha-1)\zeta}).$$

6. Відображення смуги $-\pi < \operatorname{Im}(\zeta) < \pi$ на дифузор

Шуканий багатокутник представлений на рис. 7. Він являє собою чотирикутник з двома вершинами у нескінченості. Використовуючи принцип симетрії, будемо розглядувати тільки верхню його половину – трикутник $A_1 A_2 A_3$ з кутами $\alpha_1 = 0, \alpha_2 = -\alpha, \alpha_3 = 1 + \alpha$ ($\sum \alpha_k = 1$). Назначимо точки вісі ξ , що відповідають вершинам, так: $a_1 = 0, a_2 = \infty, a_3 = -1$; враховуючи відповідність точок a_3 і A_3 , маємо:

$$z = C \int_{-1}^{\zeta} \frac{(\zeta+1)^\alpha}{\zeta} d\zeta + ih. \quad (6.1)$$

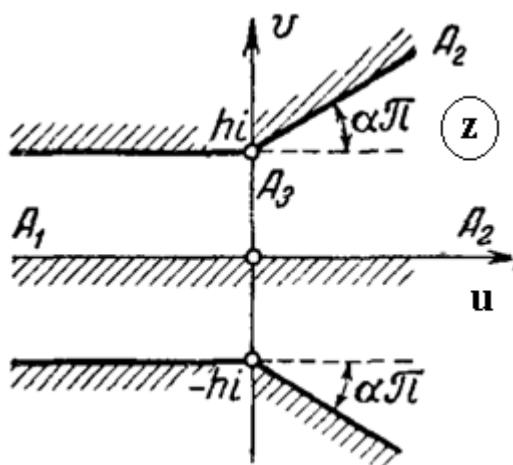


Рис. 7

Для знаходження константи С ми використаємо те, що при обході точкою ζ півколо c_r : $\zeta = re^{i\varphi}$ (φ змінюється від π до 0) функція, що визначається останнім інтегралом, отримує приріст

$$\Delta z = C \int_{c_r} \frac{d\zeta}{\zeta} + O(r) = -C\pi i + O(r)$$

(функція $(\zeta + 1)^\alpha$ на колі c_r мало відрізняється від 1: $(\zeta + 1)^\alpha = 1 + O(r)$). З іншого боку, при цьому обході відповідна точка z переходить з променя $A_1 A_3$ на промінь $A_1 A_2$, отже, Δz мало відрізняється від $-hi$. Таким чином, $C = h/\pi$, і функція, що здійснює конформне відображення верхньої напівплощини на верхню половину багатокутника на (рис. 7) має вигляд:

$$z = \frac{h}{\pi} \int_{-1}^{\zeta} \frac{(\zeta+1)^\alpha}{\zeta} d\zeta + ih. \quad (6.2)$$

Замінюючи тут ζ на e^ζ отримаємо відображення полоси $0 < \eta < \pi$ на верхню половину багатокутника (рис. 7):

$$z = \frac{h}{\pi} \left\{ \int_{i\pi}^{\zeta} (e^\zeta + 1)^\alpha d\zeta + i\pi \right\}. \quad (6.3)$$

Та ж сама функція здійснює конформне відображення полоси $-\pi < \eta < \pi$ на весь багатокутник.

В загальному випадку отриманий інтеграл виражається через гіпергеометричну функцію Гаусса. Роблячи заміну $e^\zeta = t$ цей інтеграл можна привести до інтеграла від біноміального диференціала, який за допомогою підстановок Чебишева зводиться до інтегралу, який виражається через елементарні функції у випадку раціональних α . Через складність цього інтегралу, в програмній реалізації були введені лише окремі випадки α .

7. Відображення напівплощини $\operatorname{Im}(\zeta) > 0$ на напівплощину $\operatorname{Im}(z) > 0$ з відкинутим рівнобедреним трикутником

Шуканий багатокутник представлений на рис. 8. Він являє собою чотирикутник з вершиною у нескінченності. Виходячи з властивості інтеграла Шварца-Кристоффеля – якщо вершині багатокутника відповідає нескінченно віддалена точка, то множник, що відноситься до цієї вершини, випадає – будемо розглядувати трикутник з кутами $\alpha\pi, \alpha\pi, 3\pi - 2\alpha\pi$. З геометричних співвідношень:

$$\alpha = 1 - \frac{1}{\pi} \operatorname{arctg} \frac{2h}{a}. \quad (7.1)$$

Назначимо точки вісі ξ , що відповідають вершинам трикутника, $a_1 = -1, a_2 = 0, a_3 = 1$. З формул Шварца-Кристоффеля маємо:

$$z = C \int_0^\zeta (\zeta - 1)^{\alpha-1} (\zeta + 1)^{\alpha-1} \zeta^{2-2\alpha} d\zeta. \quad (7.2)$$

Для значення цього інтегралу був отриманий вираз, що має в собі спеціальні функції:

$$z = C \left[\zeta {}_2F_1\left(-\frac{1}{2}, 1 - \alpha, \frac{1}{2}, \frac{1}{\zeta^2}\right) + \frac{i\sqrt{\pi}\Gamma(\alpha)\operatorname{tg}\alpha\pi}{\Gamma(-\frac{1}{2}+\alpha)} \right], \quad (7.3)$$

де Γ – гамма-функція, ${}_2F_1$ – гіпергеометрична функція.

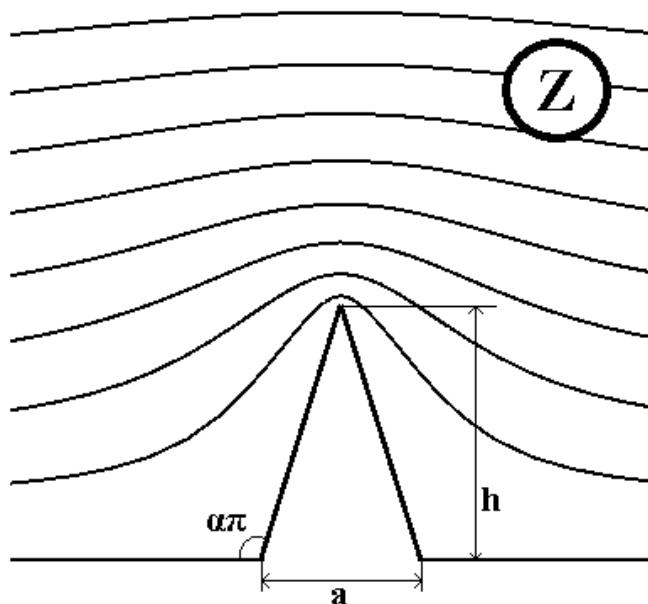


рис.8

Для знаходження невідомої константи С ми використаємо той факт, що при русі від точки $\zeta = -1$ до точки $\zeta = 1$ в допоміжній площині, ми здійснюємо рух по двох ребрах трикутника. Отже:

$$L = C \int_{-1}^1 (\zeta^2 - 1)^{\alpha-1} \zeta^{2-2\alpha} d\zeta = \frac{2C\Gamma\left(\frac{3}{2}-\alpha\right)\Gamma(\alpha)\cos\alpha\pi}{\sqrt{\pi}};$$

$$L = \frac{a}{\cos\alpha\pi};$$

$$C = -\frac{L\sqrt{\pi}\sec\alpha\pi}{2\Gamma\left(\frac{3}{2}-\alpha\right)\Gamma(\alpha)}.$$

Таким чином, шукана функція $z = z(\zeta)$ має вигляд:

$$z = -\frac{a\sqrt{\pi}}{2\Gamma\left(\frac{3}{2}-\alpha\right)\Gamma(\alpha)\cos^2\alpha\pi} \left[\zeta {}_2F_1\left(-\frac{1}{2}, 1-\alpha, \frac{1}{2}, \frac{1}{\zeta^2}\right) + \frac{i\sqrt{\pi}\Gamma(\alpha)\operatorname{tg}\alpha\pi}{\Gamma\left(-\frac{1}{2}+\alpha\right)} \right]. \quad (7.4)$$

Обчислення гамма-функції відбувається за допомогою обчислення логарифму гамма-функції, для якого є асимптотична формула (ряд Стирлінга):

$$\ln\Gamma(z) = \left(z - \frac{1}{2}\right)\ln z - z + \frac{1}{2}\ln 2\pi + \sum_{n=1}^N \frac{B_{2n}}{2n(2n-1)z^{2n-1}} + O(z^{-2N-1}),$$

де B_n – числа Бернуллі. Використовується також те, що $\overline{\ln\Gamma(z)} = \ln\Gamma(\bar{z})$, якщо $|\operatorname{arg}z| > \pi$.

Гіпергеометрична функція може бути визначена за допомогою ряду Гауса:

$$F(a, b; c; z) = {}_2F_1(a, b; c; z) = \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(c)_n} \frac{z^n}{n!} = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{n=0}^{\infty} \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)} \frac{z^n}{n!},$$

який збігається абсолютно при $|z| < 1$, а для інших випадків має аналітичне продовження.

8. Чисельний метод знаходження функції струму в задачі обтікання тіла безвихровим потоком ідеальної нестисливої рідини

При безвихровій течії ідеальної нестисливої рідини навколо нерухомого твердого тіла функція струму задовольняє рівнянню Лапласа:

$$\Delta\psi(x, y) \equiv \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0, \quad (8.1)$$

та наступним граничним умовам:

$\psi = V_y y - V_x x$ у нескінченно віддаленій точці;

$\psi = C$ на контурі тіла.

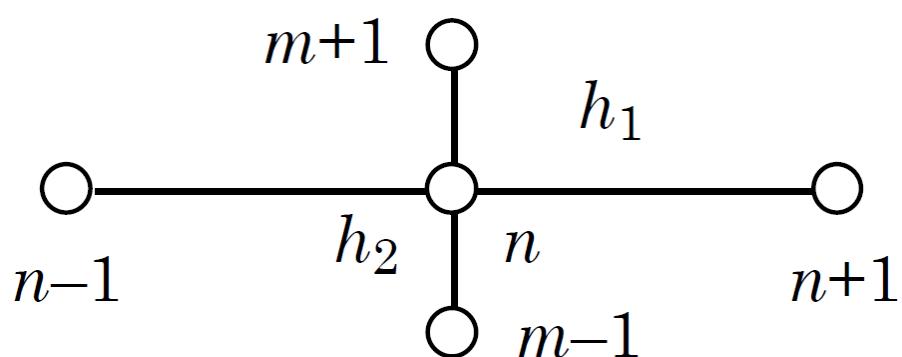
В процесі розв'язання константу C можна обрати рівною нулю. Умову $\psi = V_y y - V_x x$ можна використовувати також як початкове наближення для чисельного розв'язку.

Виберемо в області $g = [0 < x < a_1, 0 < y < a_2]$, в якій розглядується течія, прямокутну сітку, причому для спрощення – рівномірну за обома координатами:

$$x_n = nh_1 \quad (n = \overline{0, N}, h_1 = \frac{a_1}{N});$$

$$y_m = mh_2 \quad (m = \overline{0, M}, h_2 = \frac{a_2}{M}).$$

Візьмемо наступний шаблон в околі точки (x_n, y_m) :



та замінимо похідні в рівнянні Лапласа різницевими співвідношеннями:

$$\frac{\partial^2 \psi}{\partial x^2}_{n,m} = \frac{\psi_{n-1,m} - 2\psi_{n,m} + \psi_{n+1,m}}{h_1^2} + O(h_1^2);$$

$$\frac{\partial^2 \psi}{\partial y^2}_{n,m} = \frac{\psi_{n,m-1} - 2\psi_{n,m} + \psi_{n,m+1}}{h_2^2} + O(h_2^2).$$

У результаті прийдемо до системи рівнянь – різницевої схеми:

$$(y_{n-1,m} - 2y_{n,m} + y_{n+1,m})h_1^{-2} + (y_{n,m-1} - 2y_{n,m} + y_{n,m+1})h_2^{-2} = 0,$$

де $\psi(x_n, y_m) \approx y_{n,m}$. Також потрібно врахувати задані граничні умови.

За великого числа невідомих одержану систему розв'язують ітераційними методами, наприклад, за допомогою процесу Лібмана, який має таку розрахункову формулу:

$$y_{n,m}^{(s+1)} = \frac{[y_{n-1,m} + y_{n+1,m} + (y_{n,m-1} + y_{n,m+1})\alpha^2]^{(s)}}{[2(1+\alpha^2)]}, \left(\alpha = \frac{h_1}{h_2}; s = 0, 1, 2, \dots \right). \quad (8.2)$$

Ця різницева схема абсолютно (безумовно) стійка.

9. Опис програми

Програма, яка дозволяє візуалізувати отримані результати, написана на платформі .NET Framework 4.6.2 з використанням технології створення клієнтських програм Windows Presentation Foundation на мові програмування C# 6.0. Для графічних побудов була використана безкоштовна бібліотека OxyPlot.

При створенні програми автор дипломної роботи переслідував ціль створення програмного забезпечення, яке б дозволило візуалізувати конформні відображення, базуючись тільки на визначеному в програмі типі даних комплексного числа, комплексного потенціалу, а також використанні загального інтерфейсу для функцій конформного відображення, що дозволило мати єдине інтерфейсне посилання для всіх типів конформних відображень. Програма надає користувачу доступ до різноманітних налаштувань, що стосуються як загальних налаштувань (товщина та колір ліній струму, стрілки, колір заливки межі області тощо), так і конкретних відображень, наприклад при відображені напівплощини у напівплощину з відкинутим відрізком є можливість налаштувати положення та висоту відрізка. Користувачу надана можливість зберігати отримані результати у трьох популярних для Windows-систем форматах (.bmp, .png, .jpg) із можливістю регулювання розміру зображення зі збереженням вихідних пропорцій. Програма має в собі вікно довідки, зручність якого полягає в тому, що воно загальне для всієї програми і в якості вихідного файлу документа використовує документ Microsoft Office Word, який можна зручно редагувати. Для виводу документу Microsoft Office Word використовується WPF компонент DocumentViewer, що має вбудований функціонал для масштабування, пошуку слів та друку вихідного документу. Okremo від трьох розділів, в яких проходить побудова ліній течії завдяки конформним відображенням, стоїть четвертий, в якому будується ліній струму в вигляді, подібному до вигляду ліній рівня. В цьому розділу чисельно розв'язується задача Діріхле для рівняння Лапласа, якому задовольняє функція струму. Цей розділ унікальний також тому, що не має доступу до меню налаштувань.

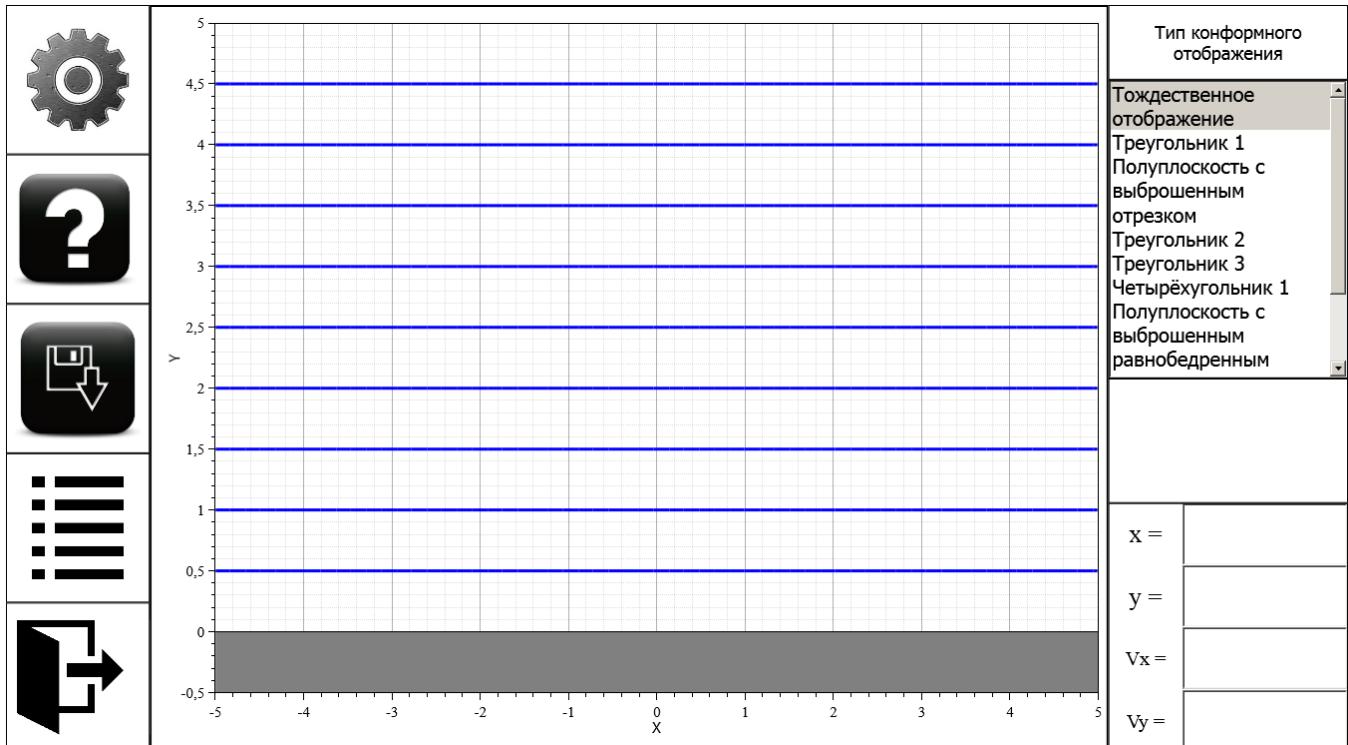
Програма складається з наступних частин:

9.1. Головне меню



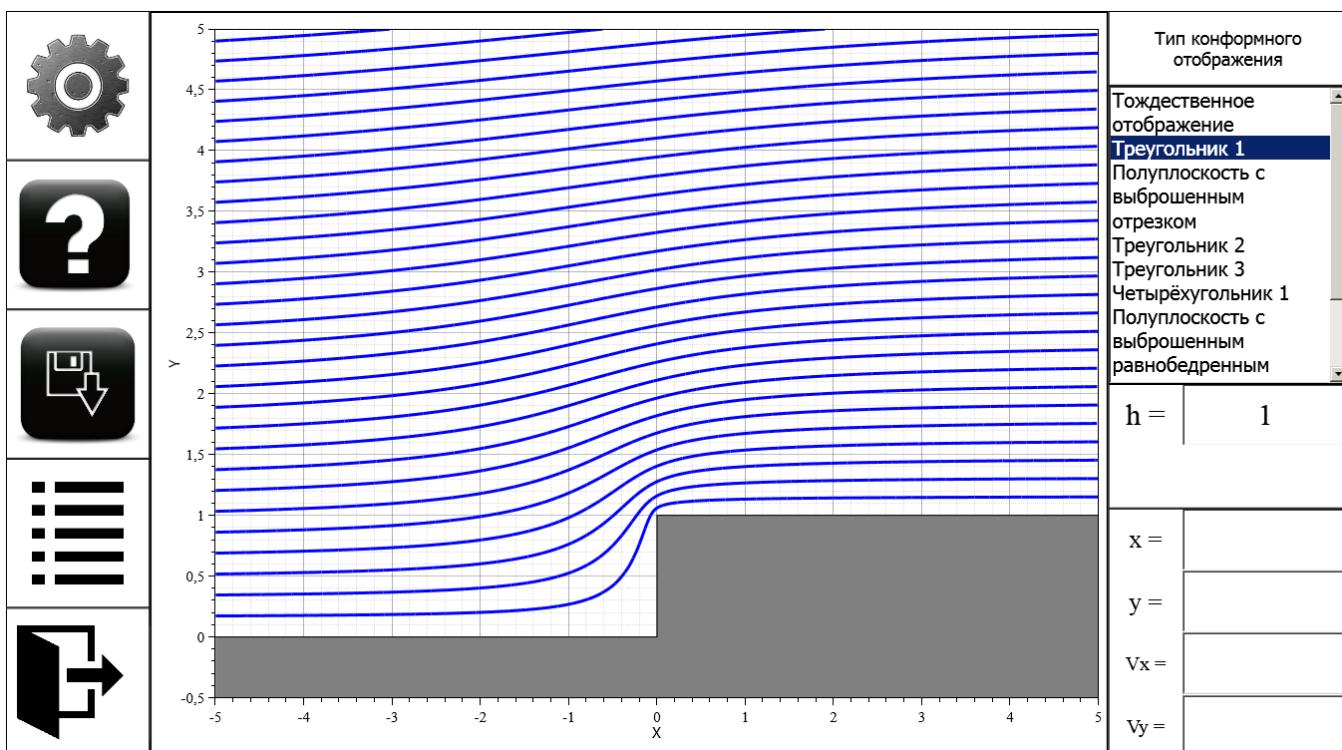
Головне меню програми забезпечує навігацію в програмі по основних розділах, які відрізняються видом канонічної області (напівплощина, полоса, коло та прямокутник) та/або способом знаходження потрібних результатів, тобто ліній струму. Кожен з цих розділів далі детально описано. В лівому нижньому куту знаходиться вікно довідки, в правому нижньому куту – кнопка для виходу з програми. При наведенні курсора мишко на одне з чотирьох зображень канонічних областей на екрані з'явиться зовнішній вигляд цієї області та стислий опис розділу.

9.2. Конформні відображення напівплощини

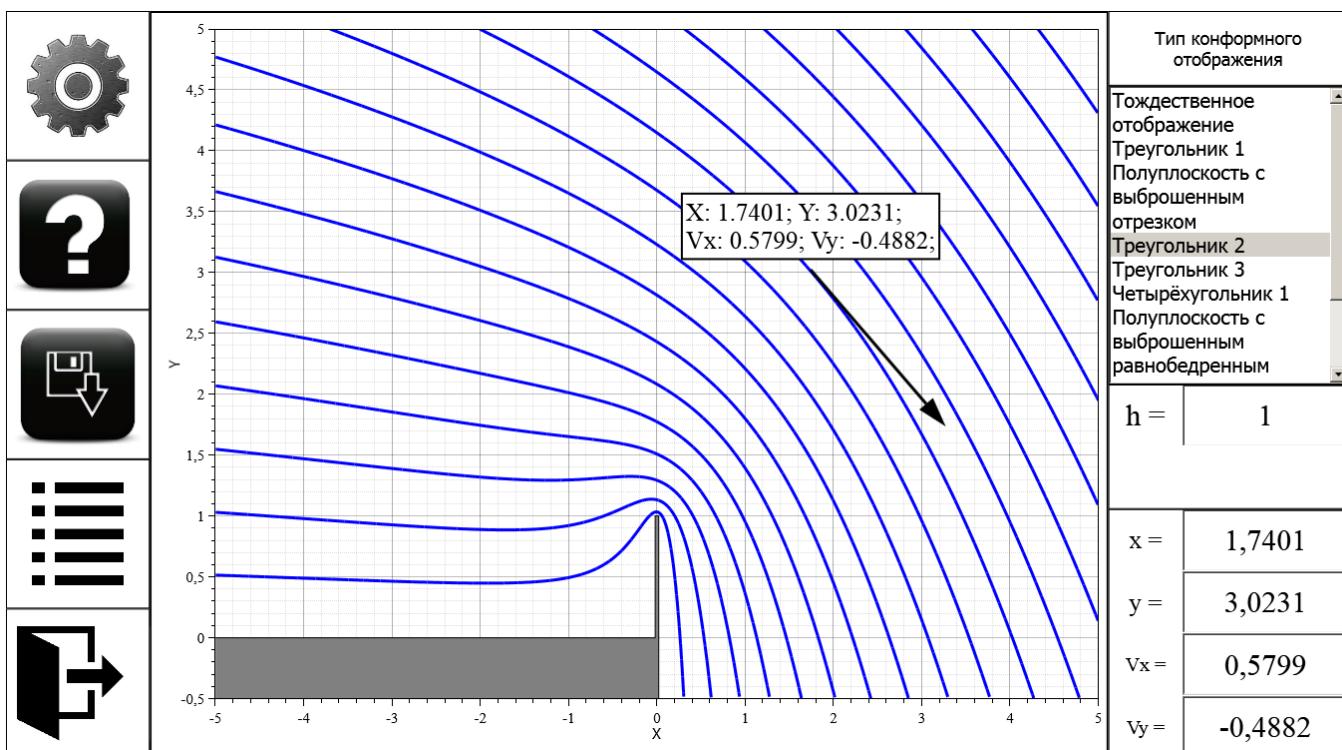


Якщо в головному меню вибрати перший (зверху) розділ зліва, то користувач попаде до розділу, в якому реалізовані конформні відображення напівплощини на різні області. Зліва знаходиться панель, через яку можна відкрити вікно налаштувань, вікно довідки, вікно зберігання отриманого результату, вихід до головного меню та вихід з програми. Справа знаходиться панель, в якій є список доступних конформних відображень, налаштування параметрів області для кожного конкретного конформного відображення, координати курсора та значення швидкостей в точці, де знаходиться курсор.

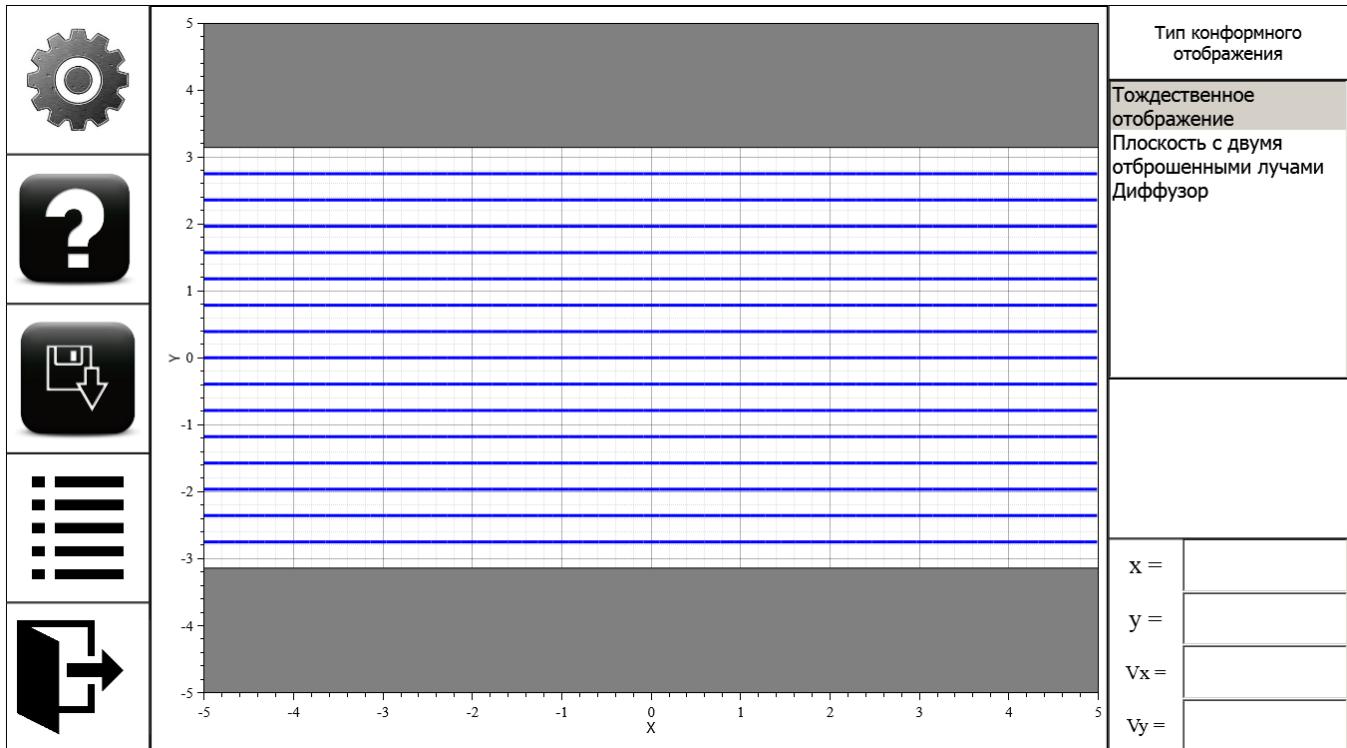
При запуску розділу одразу відбувається побудова тотожного конформного відображення, тобто такого, яке кожній точці допоміжної площини ставить у відповідність ту ж саму точку. Ці точки зберігаються в програмних списках, а потім, після вибору деякого конформного відображення, на кожну точку, яка представлена комплексним числом, діє функція конформного відображення. Після цього точка перерисовується:



Координати мишки та значення швидкостей не будуть виведені, якщо мишка виходить за межі області. Якщо курсор не вийшов за межі області, то при натискуванні на ліву кнопку мишки буде зображена стрілка, що показує напрям вектору швидкості, а також дублює інформацію з панелі виводу для координат та швидкостей:



9.3. Конформні відображення полоси $-\pi.. \pi$



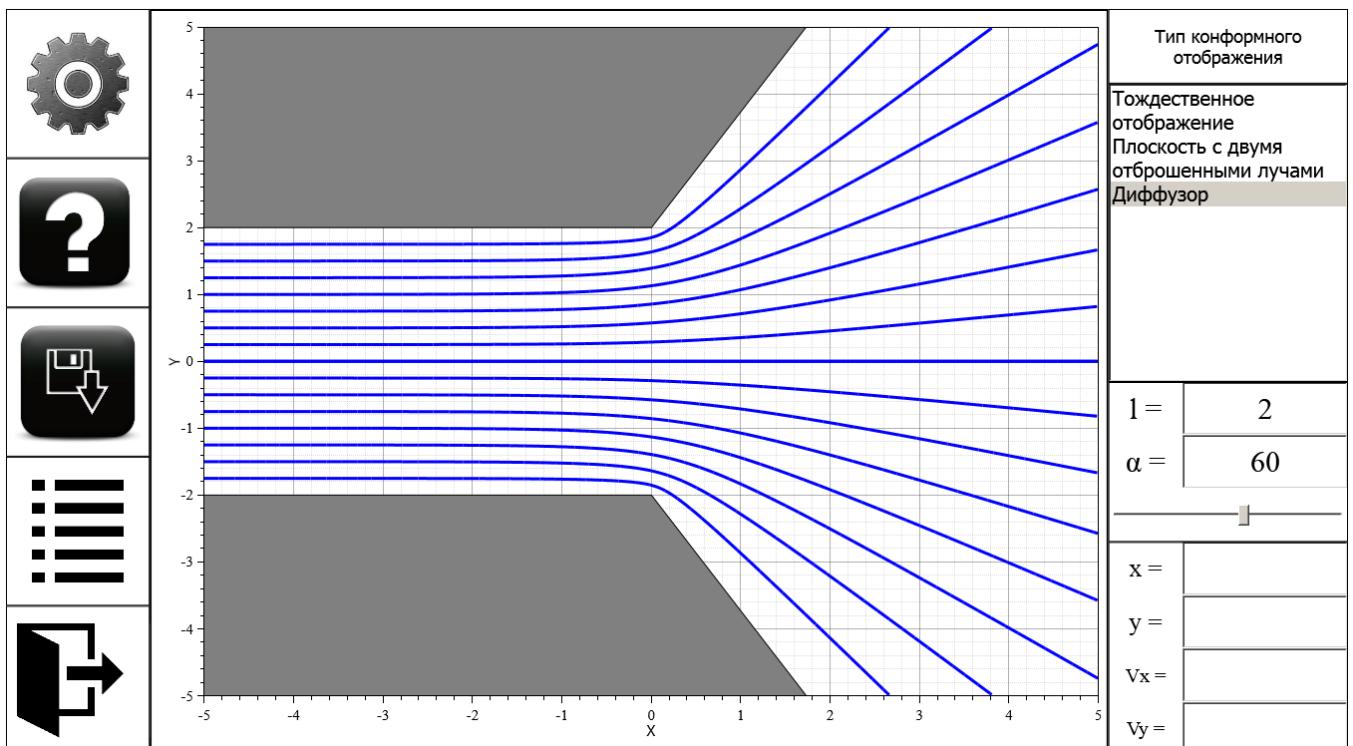
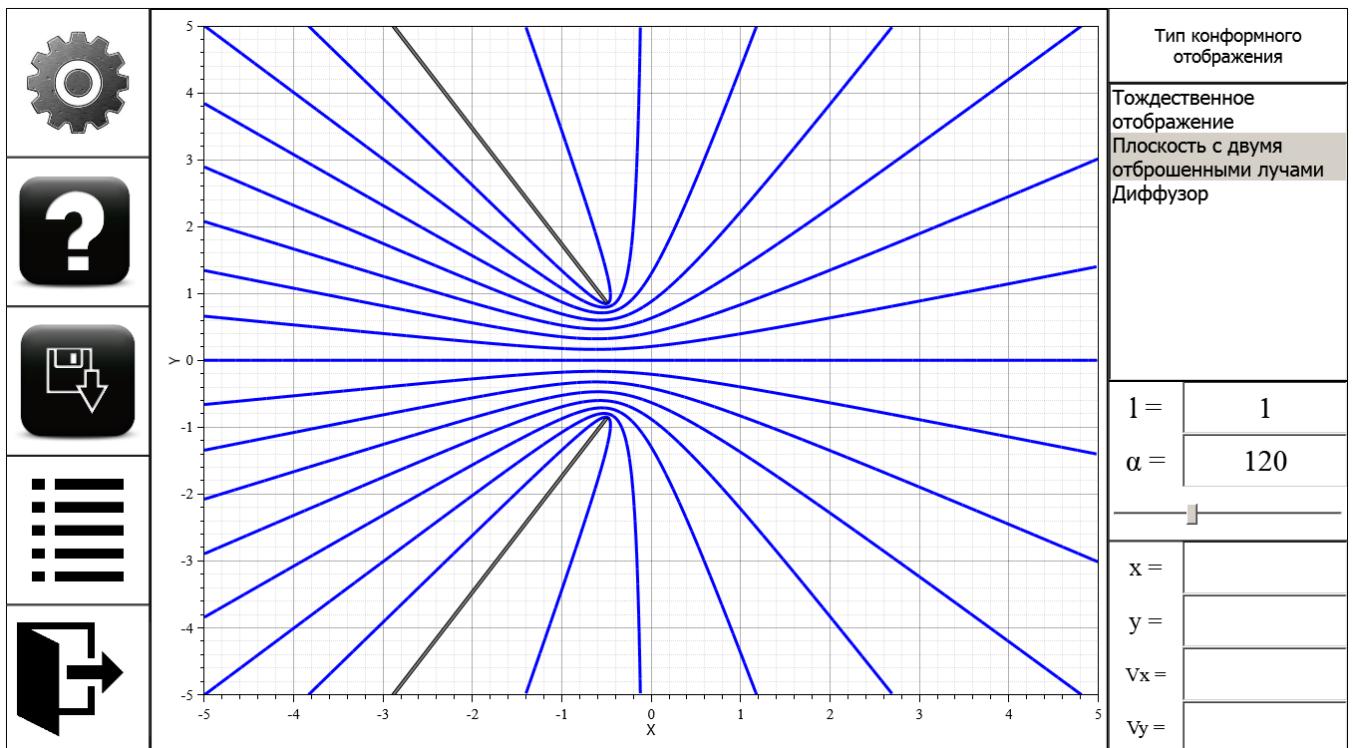
Якщо в головному меню обрати другий (зверху) розділ зліва, то можна перейти до розділу, в якому реалізовані конформні відображення області, що являє собою полосу, вертикальна координата η якої має значення від $-\pi$ до π . З точки зору екземпляра класу комплексного потенціалу, цей розділ не відрізняється від попереднього, тобто комплексний потенціал в допоміжній площині також є лінійною функцією:

$$W(\zeta) = V_\infty \zeta .$$

Тут $V_\infty = 1$.

Інтерфейс користувача такий же, як і в попередньому розділі. Зліва знаходиться панель, через яку можна відкрити вікно налаштувань, вікно довідки, вікно зберігання отриманого результату, вихід до головного меню та вихід з програми. Справа знаходиться панель, в якій є список доступних конформних відображень, налаштування параметрів області для кожного конкретного конформного відображення, координати курсора та значення швидкостей в точці, де знаходиться курсор. В цьому розділі реалізовані тільки два відображення (не враховуючи тотожне $z = \zeta$), а саме відображення полоси на площину з двома відкинутими променями та відображення полоси на дифузор. В обох випадках на панелі вводу справа з'явиться

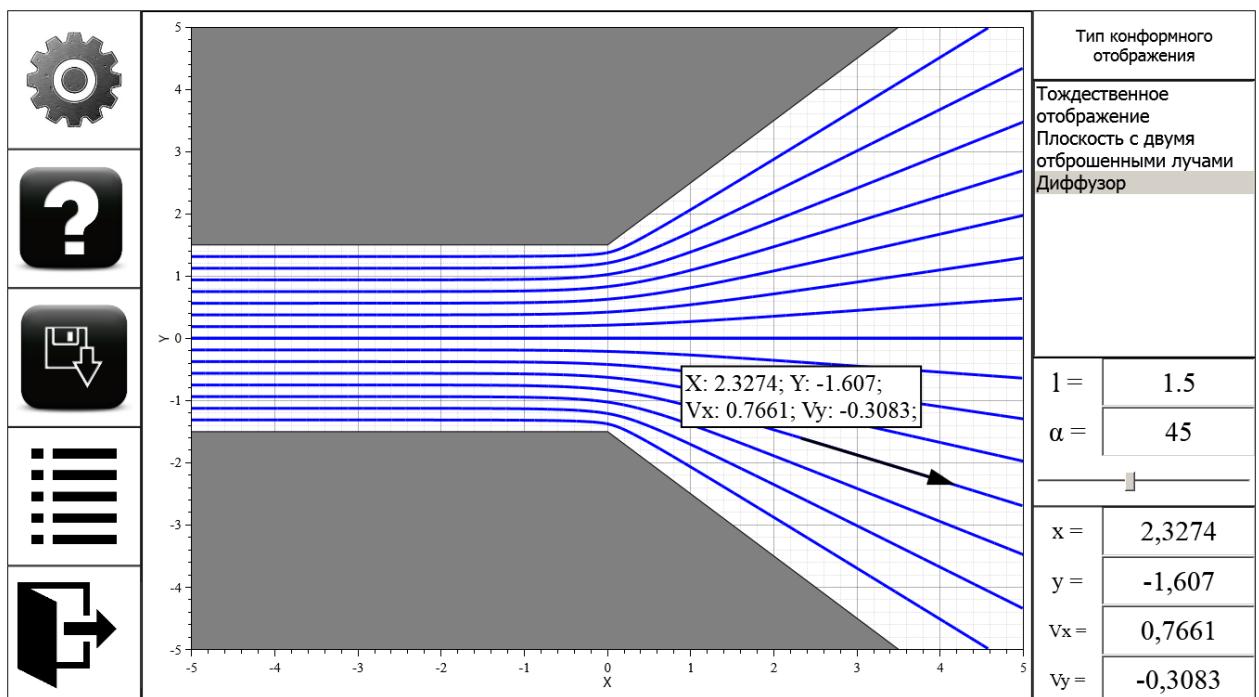
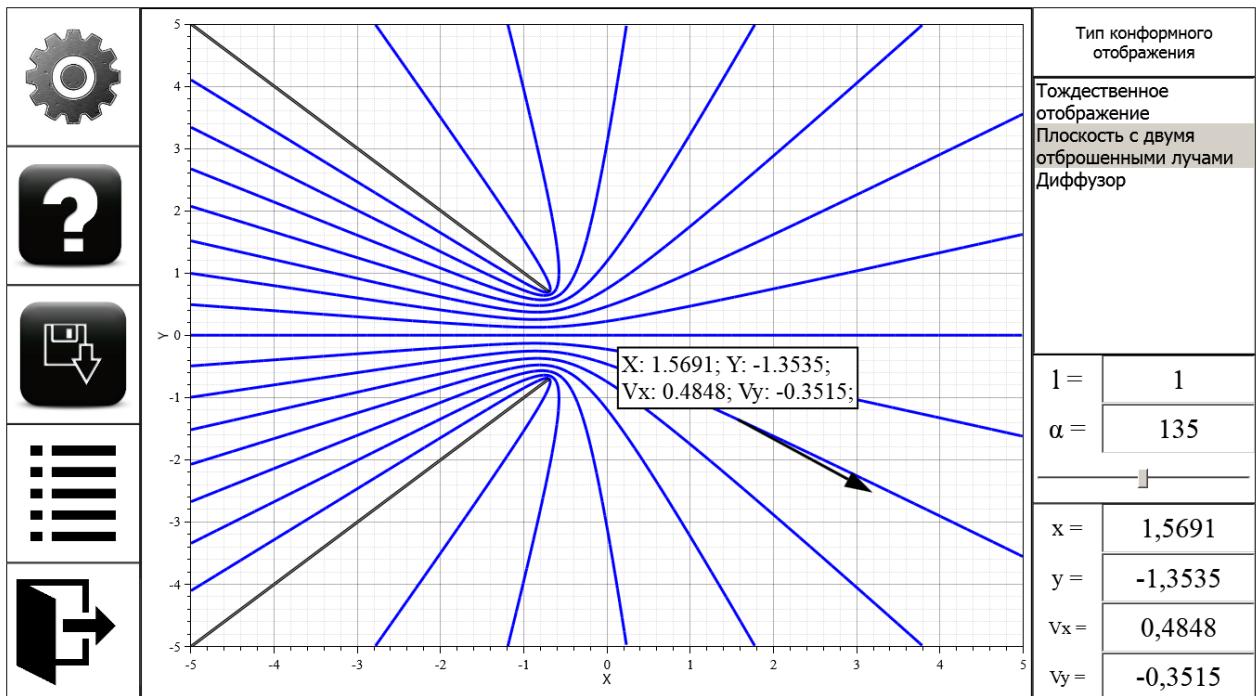
поле для вводу відстані (променів від початку координат або частин дифузору) та слайдер з полем для виводу кута:



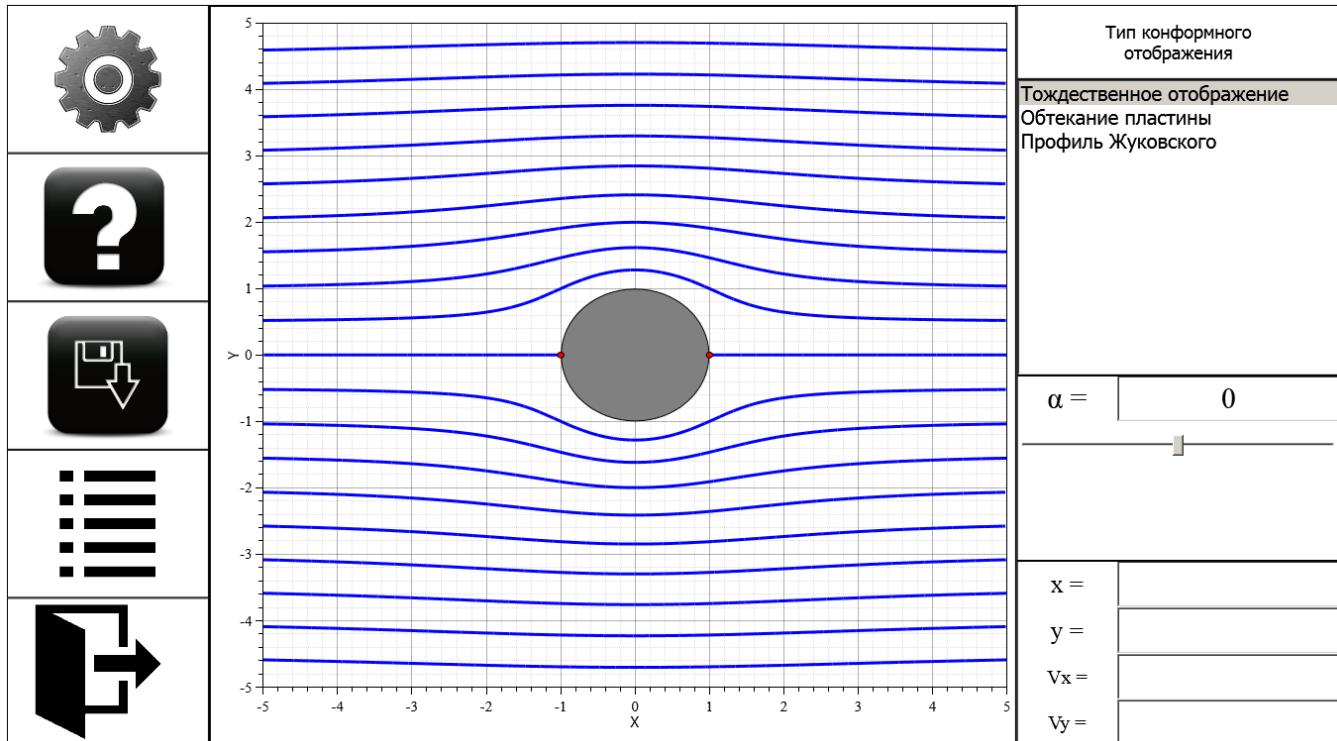
У випадку відображення полоси на площину з двома відкинутими променями кут нахилу променів може приймати значення від 90 до 180 градусів. У випадку

відображення полоси на дифузор, значення кута можуть бути: 15, 18, 22.5, 30, 36, 45, 60 та 90 градусів.

Аналогічно попередньому розділу в даному розділі в правому нижньому куту є поля виводу координат курсору мишкої та значення швидкостей в точці, де знаходиться курсор. Збережена можливість отримати зображення стрілки, що направлена в напрямку вектора швидкості, та отримати на екрані закріплена інформацію стосовно точки, де було натиснута ліва кнопка мишкої:



9.4. Конформні відображення одиничного кола

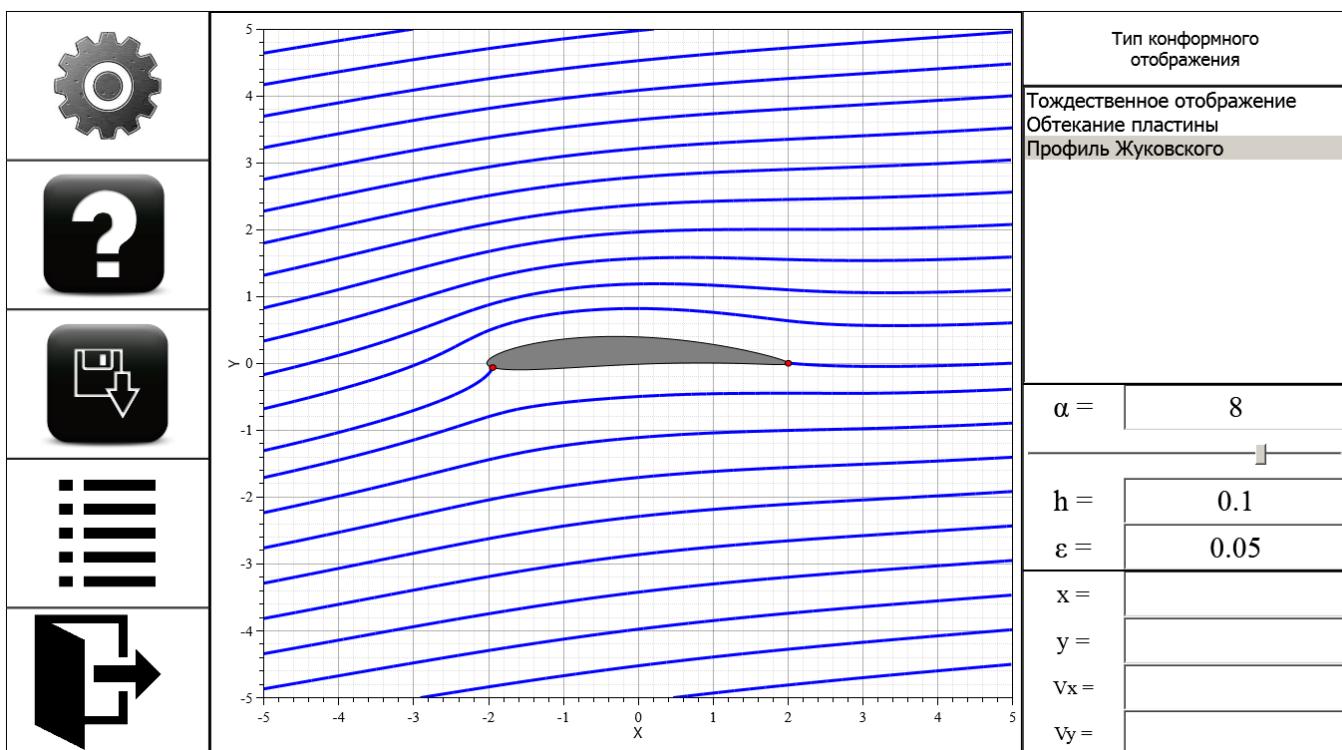


Обравши в головному меню програми перший (зверху) пункт зліва користувач отримує зображення ліній струму при обтіканні одиничного кола.

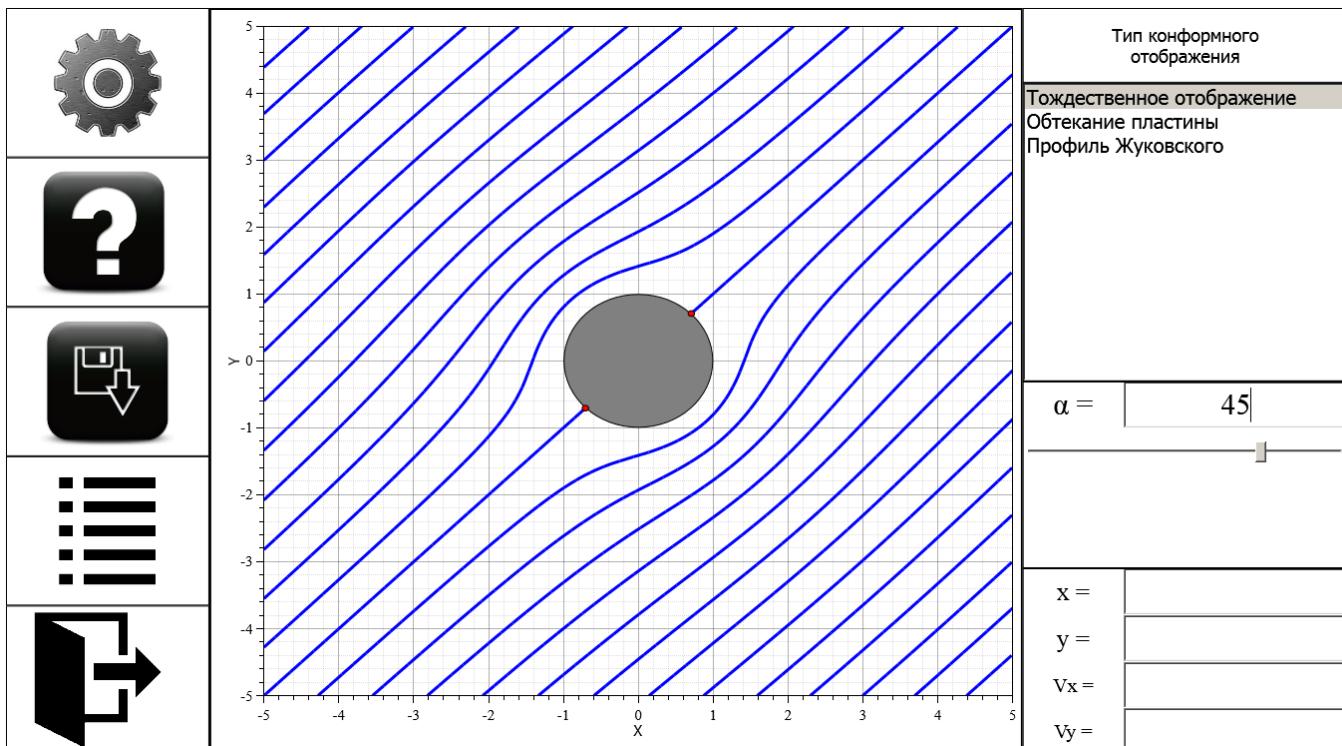
Ліва частина вікна працює аналогічно попереднім двом розділам. В правій частині постійно знаходиться слайдер, що регулює кут атаки. Кут атаки може бути від -90 до 90 градусів у випадку обтікання кола та пластини та від -15 до 15 градусів у випадку обтікання профіля Жуковського. З'явилися два нові об'єкти, а саме ліва та права точки гальмування, координати яких розраховуються з умови рівності нулю швидкостей при розв'язку задачі обтікання одиничного кола, після чого на них діє обрана функція конформного відображення.

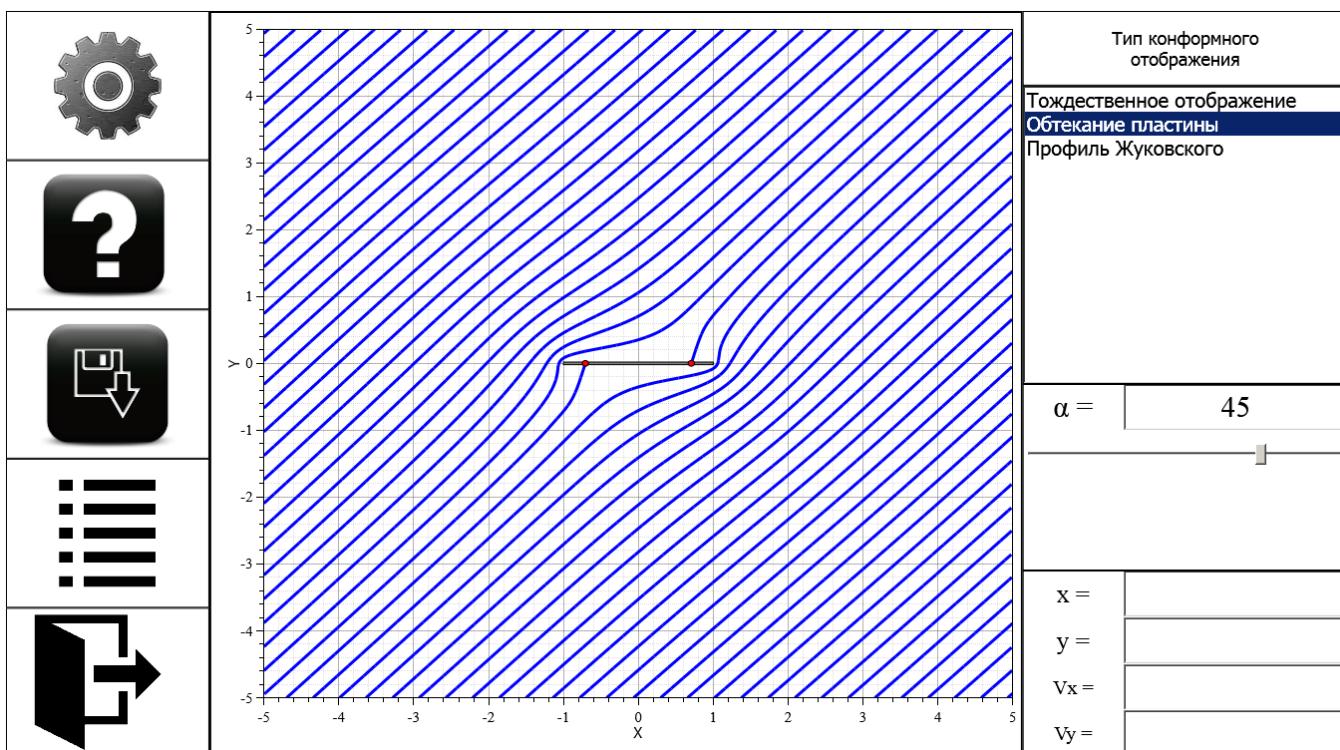
В цьому розділі швидкості у фізичній площині виводяться тільки у випадку тотожного відображення, але координати курсора мишко виводяться завжди.

Якщо в списку в правому верхньому куту обрати профіль Жуковського, під слайдером з'являться поля для налаштування профілю, а саме його кривизни та товщини:



Кут атаки, обраний при візуалізації обтікання одиничного кола, залишається таким же при обтіканні пластини (а також профілю Жуковського, якщо його значення по модулю не перевищує 15 градусів):





Слід зазначити, що даний розділ програми несе демонстраційний характер. За цієї причини функціонал розділу стосовно виводу швидкостей в точках було зменшено.

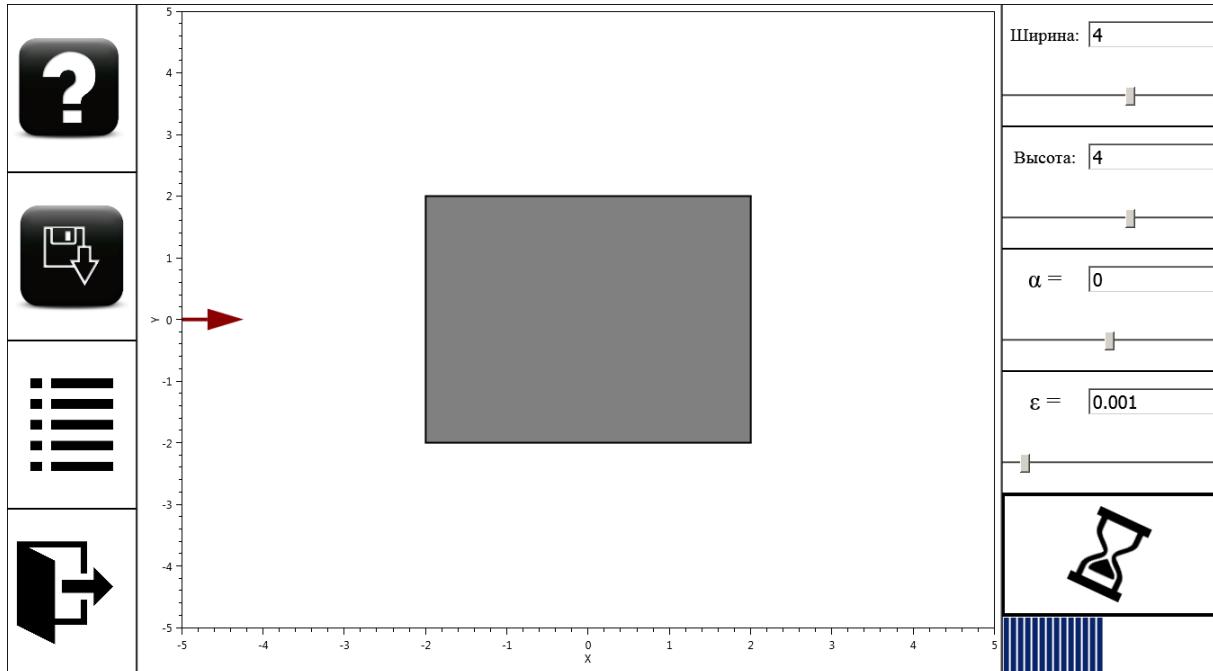
Даний розділ було добавлено в програму лише для демонстрації розширеності архітектури програми, тобто загальності застосованого підходу написання коду, що базується на класі комплексного потенціалу й використанні інтерфейсного посилання на клас конформного відображення. В даному розділі зміна в архітектурі була внесена тільки в спосіб побудови ліній струму: в цьому розділі використовується метод Рунге-Кутти 4-го порядку для розв'язання рівняння ліній струму:

$$\frac{dy}{dx} = \frac{V_y}{V_x}.$$

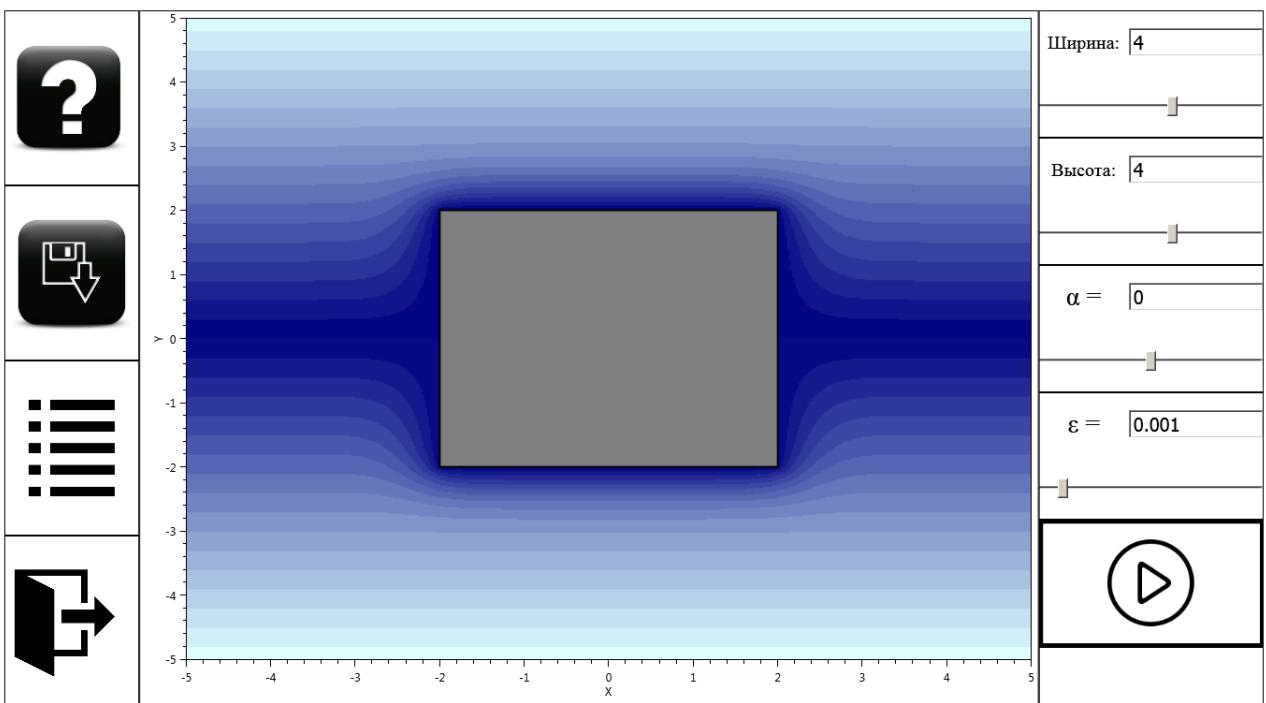
Метод Рунге-Кутти 4-го порядку використовується також для розрахунку нульової лінії струму.

При візуалізації обтікання одиничного кола та пластини в комплексному потенціалі циркуляція Γ вважається рівною нулю, а при обтіканні профілю Жуковського вона визначається завдяки постулату Чаплигіна-Жуковського.

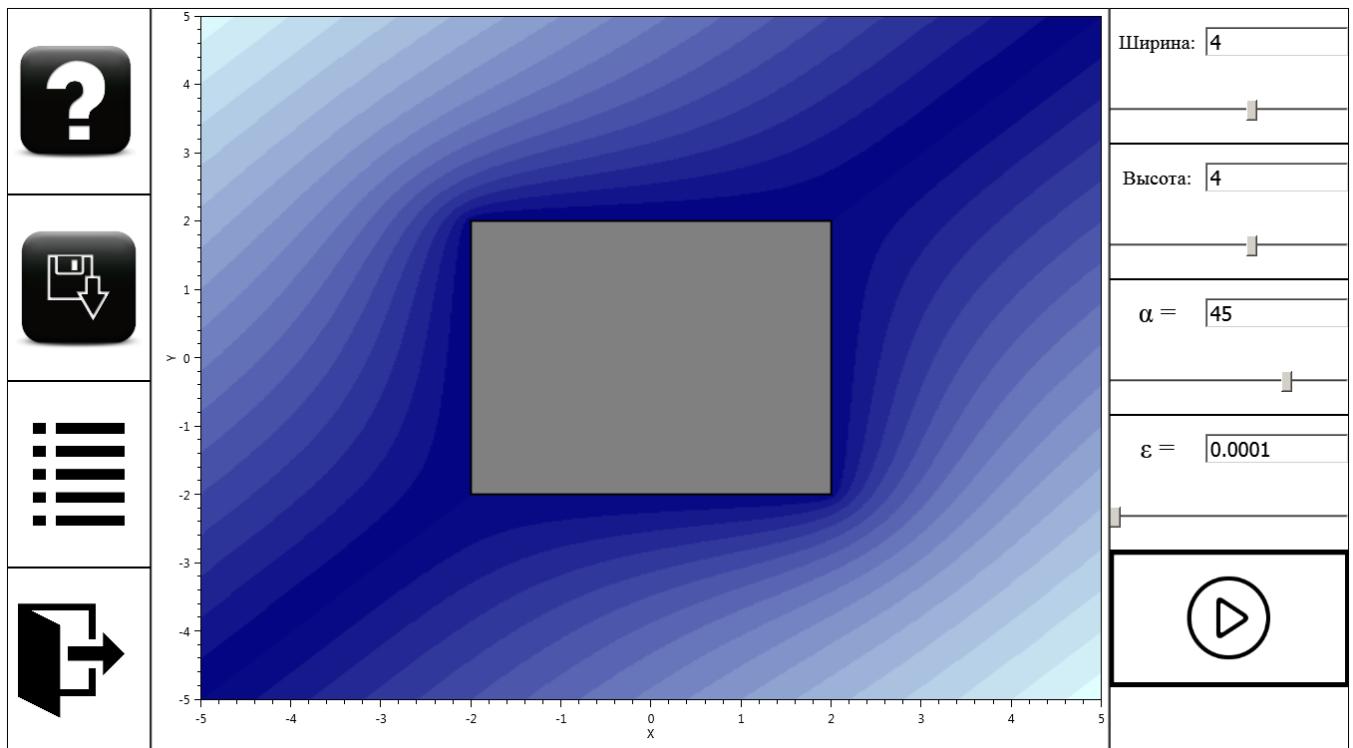
9.5. Обтікання прямокутника



Обравши в головному меню програми другий (зверху) пункт зліва користувач отримує зображення прямокутника, стрілки, яка ілюструє кут атаки, та смугу завантаження чисельного розв'язку поруч з анімованим зображенням пісочного годинника. Після того, як зображення годинника зміниться на зображення стрілки, користувач може отримати візуалізацію чисельного розв'язку задачі обтікання прямокутника:



Цей розділ відрізняється від попередніх як способом розв'язку задачі обтікання, так і зовнішнім виглядом. З лівої панелі було видалено кнопку, що відкривала меню візуальних налаштувань, на правій панелі відсутній список доступних побудов та панель виводу. Завдяки панелі справа можна тільки проводити налаштування розмірів прямокутника, значення кута атаки та точності розрахунків. Для кожного з цих параметрів є слайдер. При зміні якогось параметру проводиться ануляція отриманого розв'язку і повторний запуск методу, який займається його пошуком. Зміна розмірів прямокутника та значення кута атаки відбувається в реальному часі. Пошук розв'язку проводиться в окремому програмному потоці, що дозволяє змінити деякі параметри на панелі справа не дочекавшись завершення пошуку. В цьому окремому потоці запускається ще один програмний потік, який займається анімацією годинника, а також проводиться постійне оновлення даних смуги завантаження, що показує, скільки приблизно потрібно чекати до того моменту часу, коли буде досягнута встановлена точність. Після того, як розрахунки були завершені та була натиснута кнопка зі стрілкою, зображення червоної стрілки зникає і з'являється візуалізація даних, отриманих при чисельному розв'язку задачі Діріхле для рівняння Лапласа, якому задовольняє функція струму:



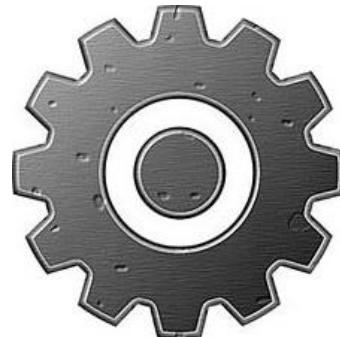
Тут кожний перехід між кольорами можна вважати окремою лінією струму.

При чисельному розв'язку задачі Діріхле для рівняння Лапласа було застосовано ітераційний процес Лібмана, виконання якого відбувається до того моменту, коли максимальний модуль різниці між розв'язками на двох сусідніх ітераціях стане меншим за встановлену точність Eps.

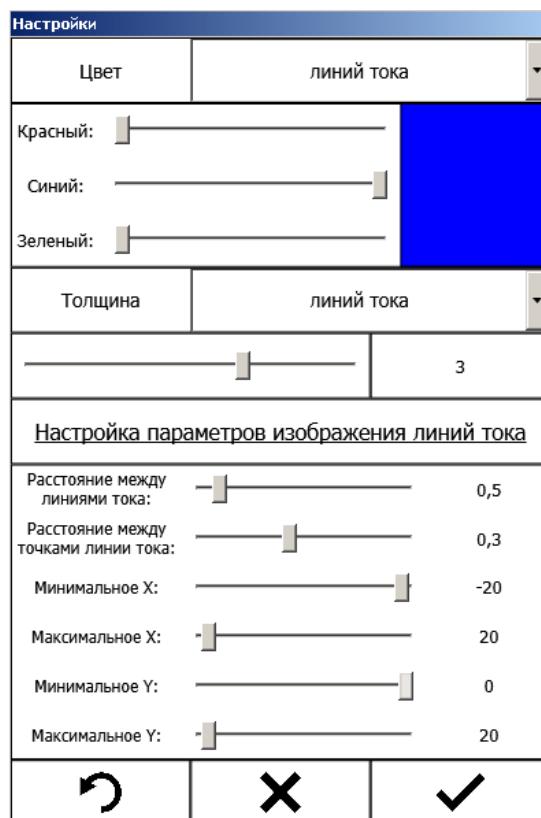
9.6. Допоміжні вікна

До допоміжних вікон належать вікно налаштувань, вікно збереження результату та вікно довідки.

Доступ до вікна налаштувань відбувається за допомогою анімованої кнопки наступного вигляду:



Вікно налаштувань виглядає наступним чином:



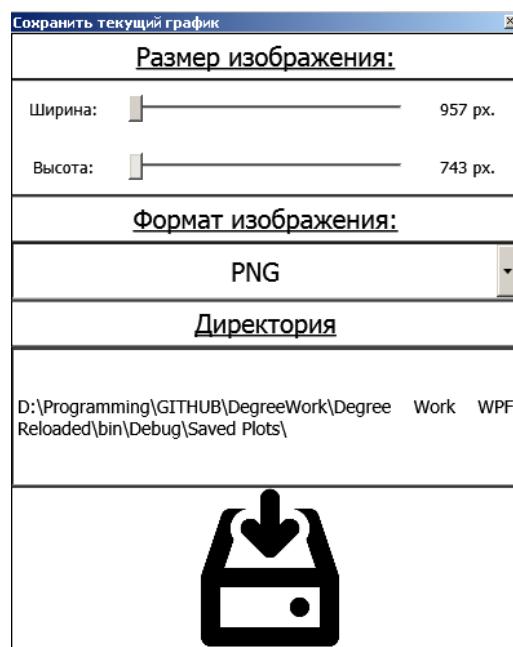
В цьому вікні можна налаштувати колір ліній струму, контуру границі області, границі та стрілки, що показує напрям вектору швидкості. Також можна налаштувати товщину вищезазначених елементів (крім границі). Налаштувати можна розміри області, в якій проводиться пошук ліній струму (при тотожному відображення, тобто у допоміжній площині), відстань між лініями струму та відстань між точками ліній струму. Це може бути доцільним, коли при застосуванні функції конформного відображення не видно всіх ліній струму. Також це корисно при візуальних дефектах при зображення ліній.

Ліва кнопка знизу встановлює вихідні значення всіх параметрів, центральна кнопка скидає значення до тих, які були збережені останніми, а права кнопка виконує збереження встановлених параметрів.

Доступ до вікна збереження відбувається за допомогою кнопки наступного вигляду:



Вікно збереження виглядає наступним чином:



Це вікно дозволяє користувачу налаштовувати розмір зображення в пікселях, задати формат зображення та директорію, куди потрібно зберегти зображення. На збереженому зображенні зберігаються лінії струму, границі області, стрілки та текст. Також враховуються зміни, які були внесені за допомогою вікна налаштувань. Налаштовувати можна ширину зображення, а висота регулюється автоматично так, щоб були збережені вихідні пропорції зображення. Зображення можна зберегти у форматах PNG, BMP та JPG. Можна задати будь-яку директорію для зображення, але спочатку автоматично обирається папка Saved Plots, що лежить в тій же самій директорії, де знаходиться файл EXE програми. При відсутності вказаної директорії, вона буде створена автоматично. Після завершення налаштувань зображення, потрібно натиснути на анімовану кнопку внизу вікна і дочекатися зміни її зовнішнього вигляду.

Вікна налаштувань та збереження зображень зроблені таким чином, що вони автоматично закриваються при втрачанні фокусу, тобто якщо користувач, наприклад, натисне ліву кнопку мишко поза ними.

Доступ до вікна довідки відбувається за допомогою кнопки наступного вигляду:



Ця кнопка присутня у всіх чотирьох розділах, а також вона є в головному меню програми. Якщо натиснути цю кнопку в головному меню програми, користувач отримує вікно наступного вигляду:

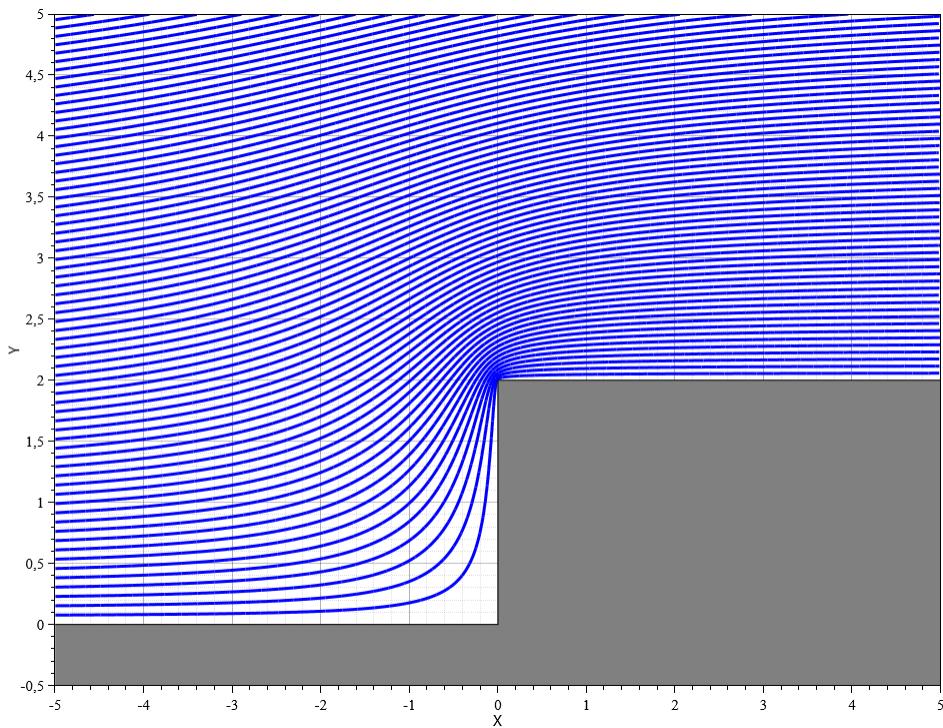
Программа выполнена в рамках дипломной работы бакалавра студента механико-математического факультета ДНУ имени Олеся Гончара Костенко Никиты Викторовича (группа МХ-13-1).

Целью написания данной программы было создание приложения, способного визуализировать различные конформные отображения, примененные к некоторым течениям с известным потенциалом, а также реализовать общий подход к решению задачи безвихревого обтекания профиля идеальной несжимаемой жидкостью, основанный на численном решении уравнения Лапласа для функции тока. Особенностью данной работы является то, что в исходном коде определены класс для потенциала течения и интерфейс конформного отображения. Все представленные в программе конформные отображения реализуют этот интерфейс, что позволяет использовать полиморфизм в классе потенциала, определив в нем интерфейсную ссылку для функции конформного отображения. Также особенностью является то, что все данные, что касаются непосредственно потенциала, так или иначе имеют представления в виде комплексного числа, исходный код и методы которого определены в другой сборке, созданной автором данной программы. Раздел программы, предназначенный для визуализации обтекания прямоугольника является независимым от других частей в том плане, что в нём не используются вышеупомянутые типы, логика которых обусловлена теорией функции комплексного переменного.

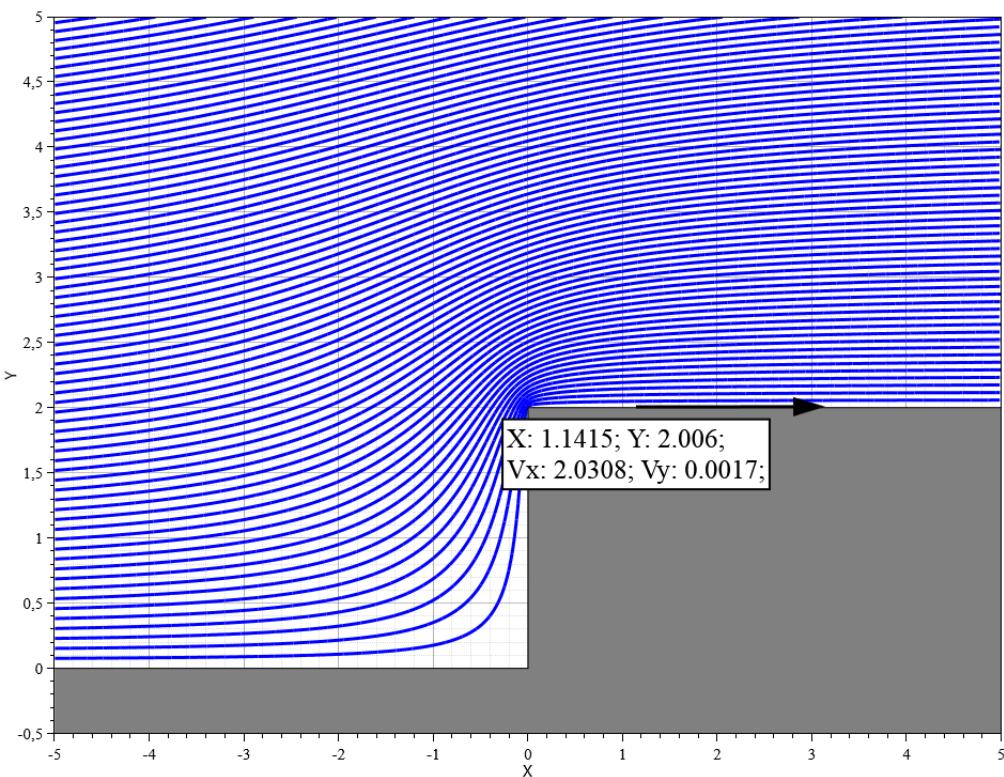
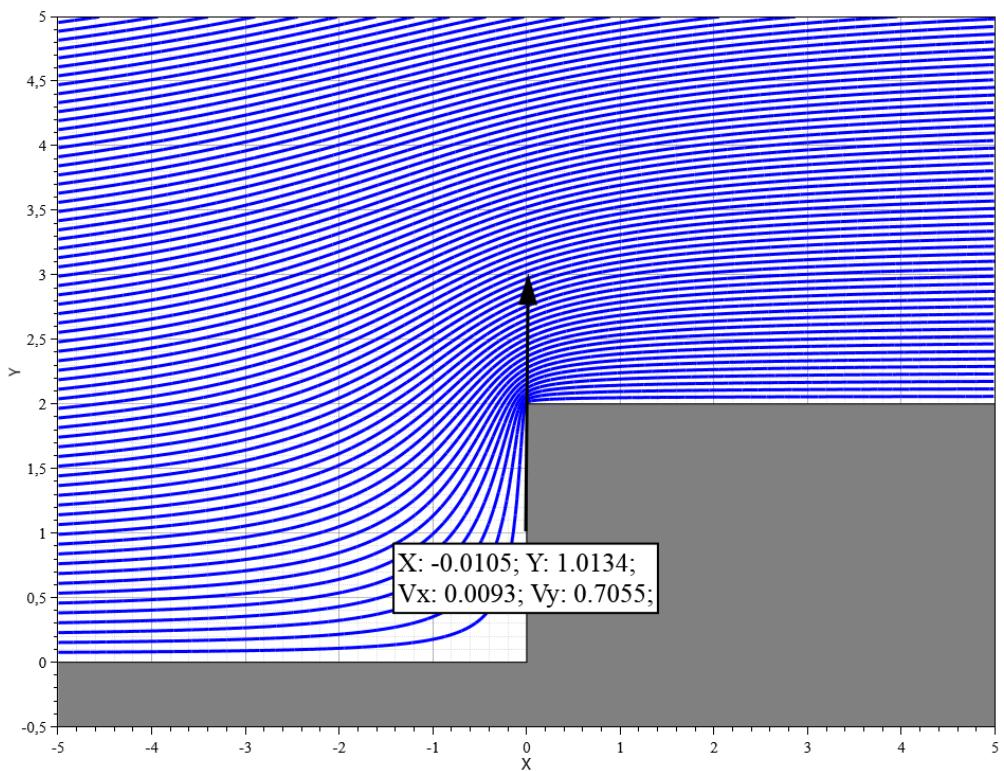
Вікно моє компонент DocumentViewer, що здатне виводити на екран вміст документів Microsoft Office Word. В списку справа відображуються назви доступних документів. Документ, який з'являється при запуску вікна залежить від того, з якого розділу воно було відкрито, проте користувач власноруч може вибрати будь-який інший документ. Якщо документ не буде знайдено в директорії для ресурсів проекту, буде виведено повідомлення про помилку. Компонент DocumentViewer здатний також проводити масштабування документа, друкувати його на принтері та проводити пошук слів в документі.

10. Обговорення отриманих результатів

Виберемо розділ, в якому реалізовані конформні відображення напівплощини $Im\zeta > 0$ та виберемо в списку другий пункт, що являє собою відображення напівплощини у напівплощину зі сходинкою. Також проведемо налаштування, щоб збільшити кількість ліній струму на екрані:

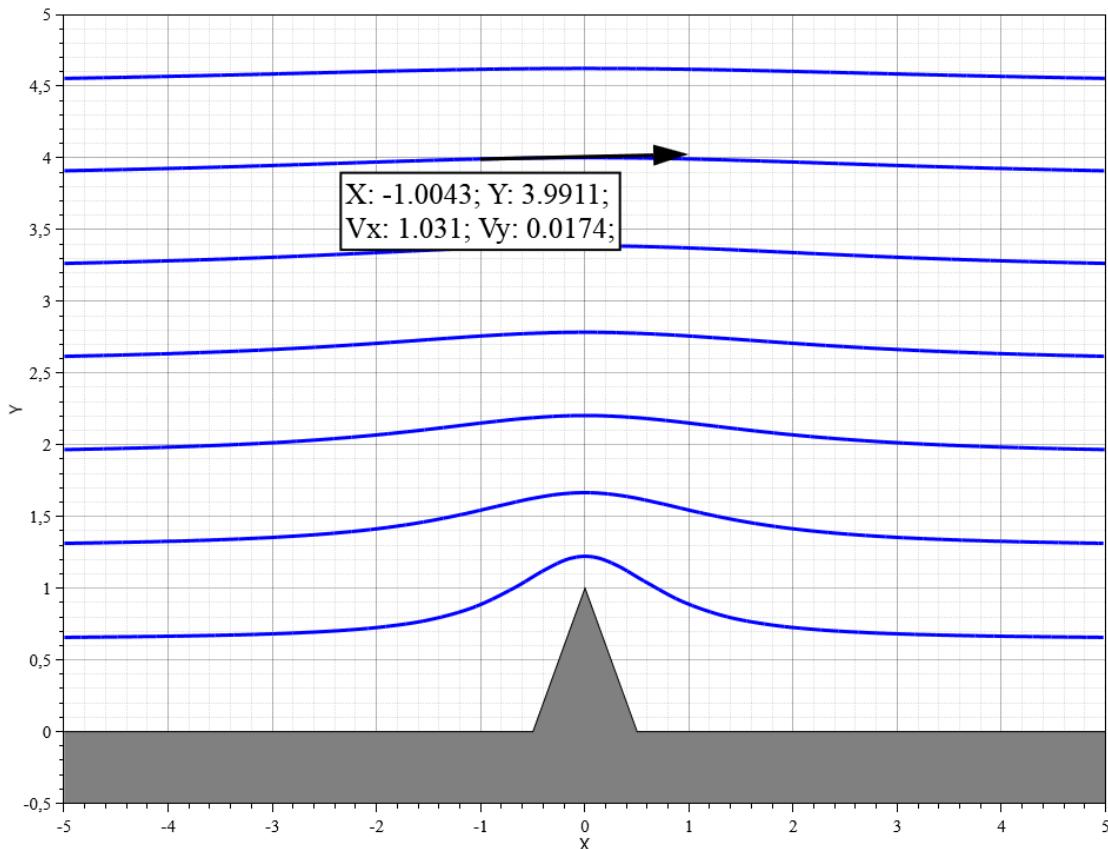


Для перевірки результату можна натиснути на деяку точку зліва та зверху від сходинки та переконатись у виконанні умови непротікання рідини:

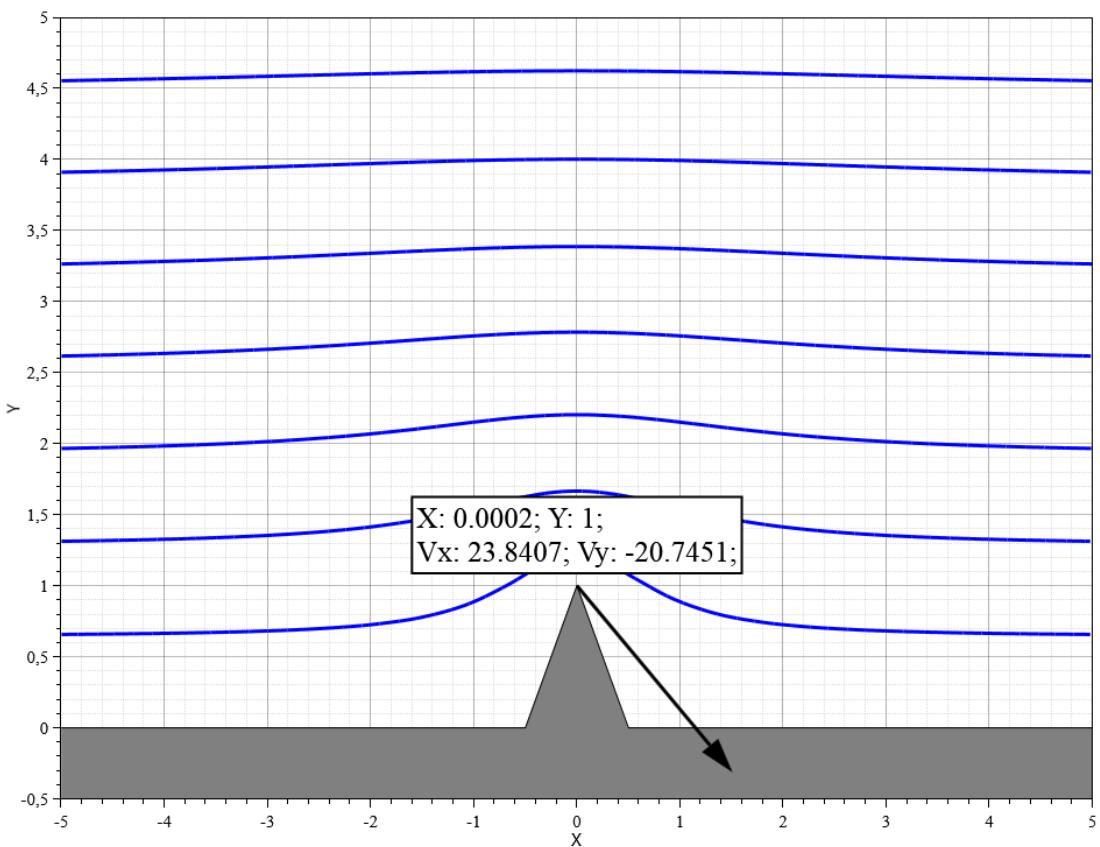


Видно, що числові значення горизонтальної компоненти швидкості на першому малюнку та вертикальної компоненти на другому набагато менші за значення вертикальної компоненти швидкості на першому малюнку та горизонтальної компоненти швидкості на другому, що свідчить про виконання умови непротікання рідини.

Конформні відображення, які здійснюються за допомогою інтеграла Шварца-Кристоффеля особливі тим, що швидкість в кутових точках у фізичній площині може прямувати до нескінченності. Перевіримо це на прикладі відображення напівплощини у напівплощину з відкинутим рівнобедреним трикутником, у якого основа та висота дорівнюють 1. Виберемо це конформне відображення в першому розділі та натиснемо на якусь точку в області за допомогою лівої кнопки мишко:



Збільшимо зображення області біля вершини трикутника, натиснемо лівою кнопкою мишко на точку біля вершини та повернемо вихідний розмір зображення:



Видно, що значення швидкостей біля вершини дуже збільшились порівняно зі значеннями на попередньому малюнку. При подальшому наближенні до вершини значення компонент швидкості будуть і далі збільшуватися до тих пір, поки не стануть рівними нескінченності (на панель елементів справа знизу нічого не буде виведено, а також буде прибрана можливість поставити стрілку на екрані).

Висновки

При виконанні даної дипломної роботи були освоєні методи дослідження безвихрового руху ідеальної нестисливої рідини при обтікання тіл у вигляді багатокутників. Перший метод базується на використанні інтегралу Шварца-Кристоффеля для побудови функцій конформного відображення таких областей, комплексний потенціал яких можна легко побудувати, на області, що являють собою узагальнені багатокутники. Була створена програма, за допомогою якої можна візуалізувати конформні відображення областей з відомим комплексним потенціалом на узагальнені багатокутники, проводити візуальні налаштування, аналізувати течію ідеальної нестисливої рідини в окремих точках та зберігати результат у вигляді зображення на електронному носії інформації. Вихідний код програми має тип комплексного потенціалу, який має методи, за допомогою яких можна визначити значення комплексного потенціалу та швидкостей в різних точках. Цей тип також має в собі інтерфейсне посилання на тип, який успадковує інтерфейс функції конформного відображення. Отже, маючи лише одне таке посилання, екземпляр типу комплексного потенціалу може працювати з усіма конформними відображеннями, для яких є визначений тип і які успадковують вищезазначений інтерфейс. Програма була розроблена таким чином, що можна легко додавати нові конформні відображення, формально для цього потрібно лише визначити тип нового конформного відображення та успадкувати інтерфейс для функцій конформного відображення.

Інший метод дослідження обтікання багатокутників, який був досліджений в даній роботі та реалізований у програмі, полягає в чисельному розв'язанні задачі Діріхле для рівняння Лапласа, якому задовольняє функція струму. Чисельний розв'язок цієї задачі знаходився за допомогою процесу Лібмана. Цей підхід ефективний для візуального представлення руху рідини, але він менш інформативний ніж підхід, що базується на використанні конформних відображень. Мається на увазі те, що користувач не отримує значення швидкостей в усіх точках (для цього потрібно було б розв'язувати ту ж саму задачу Діріхле для рівняння Лапласа, але для функції

потенціалу φ , де $\vec{V} = \text{grad}\varphi$ при безвихровому русі). Переваги цього підходу полягають в тому, що його нескладно поширити на ті багатокутники, для яких підхід, що базується на використанні інтегралу Шварца-Кристоффеля, був би набагато складнішим з точки зору розрахунків.

Слід зазначити, що отримані результати обмежені. Обмеженість використання інтегралу Шварца-Кристоффеля полягає в тому, що швидкість в кутових точках виявляється нескінченною. Обмеженість чисельного розв'язку задачі Діріхле для рівняння Лапласа, якому задовольняє функція струму, полягає в тому, що завдяки цьому можливо тільки візуалізувати безвихровий рух ідеальної рідини. В то ж час, незважаючи на обмеженість, результати, отримані за допомогою конформних відображень, можна використовувати, наприклад, для розрахунку примежового шару, приймаючи поле швидкостей, отримане в програмі, за швидкість поза шаром. Другий підхід зручно використовувати для візуалізації безвихрового руху ідеальної нестисливої рідини для областей, де використання конформних відображень завдяки інтегралу Шварца-Кристоффеля є невіправдано складним.

Список використаних джерел та літератури

- Лаврентьев М.А., Шабат Б.В. Методы теории функции комплексного переменного. –М.: Наука, 1973, –736 с. с илл.
- Кочин Н.Е., Кибель И.А., Розе Н.В. Теоретическая гидромеханика, ч. 1. –М.: Физматгиз, 1963, –584 с. с илл.
- Лойцянский Л. Г. Механика жидкости и газа. Уч. пособие для вузов. –М.: Наука, 1970, –904 с.
- Градштейн И.С. и Рыжик И.М. Таблицы интегралов сумм рядов и произведений. –М.: Физматгиз, 1963, –1100 с. с илл.
- Абрамович М., Стиган И. Справочник по специальным функциям с формулами, графиками и таблицами. –М.: Наука, 1979, –832 с с илл.
- Мусіяка В.Г. Основи чисельних методів механіки: Підручник. – К.: Вища освіта, 2004. – 240 с.: іл.
- Албахари Д., Албахари Б. C# 6.0. Справочник. Полное описание языка, 6-е изд. : Пер. с англ. – М.: ООО «И.Д. Вильямс», 2017. – 1040 с.: илл. – Парал. тит. англ.
- Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 3-е изд. – СПб.: Питер, 2012. – 928 с.: ил.
- <https://msdn.microsoft.com/ru-ru/default.aspx>
- <https://github.com/oxyplot>
- <http://www.oxyplot.org/>
- <https://metanit.com/sharp/>

Додатки

1. Лістинг кодів Wolfram Mathematica та Maple

Як було вказано на стр.28 даної дипломної роботи, інтеграл

$$z = \frac{h}{\pi} \left\{ \int_{i\pi}^{\zeta} (e^\zeta + 1)^\alpha d\zeta + i\pi \right\}$$

можна привести за допомогою заміни $e^\zeta = t$ до інтегралу від біноміального диференціала, який за допомогою підстановок Чебишева зводиться до інтегралу, який виражається через елементарні функції у випадку раціональних α . Остаточний результат та програмний код було отримано в пакеті Maple 14 наступним чином (приклад для $\alpha = \frac{\pi}{12}$):

$$\begin{aligned} & \frac{h}{\text{Pi}} \cdot \text{simplify} \left(\text{subs} \left(x = (1 + \exp(z))^{\frac{1}{12}}, 12 \cdot \text{int} \left(\frac{x^{12}}{x^{12} - 1}, x \right) \right) - \text{subs} \left(z = I \cdot \text{Pi}, \text{subs} \left(x = (1 \right. \right. \right. \\ & \left. \left. \left. + \exp(z))^{\frac{1}{12}}, 13 \cdot \text{int} \left(\frac{x^{12}}{x^{12} - 1}, x \right) \right) \right) + I \cdot \text{Pi} \right) \\ & \frac{1}{\pi} \left(h \left(-\frac{1}{12} I \pi + \ln((1 + e^z)^{1/12} - 1) - \frac{1}{2} \ln((1 + e^z)^{1/6} + (1 + e^z)^{1/12} + 1) \right. \right. \\ & \left. \left. + \frac{1}{2} \ln((1 + e^z)^{1/6} - (1 + e^z)^{1/12} + 1) - \ln((1 + e^z)^{1/12} + 1) - 2 \arctan((1 \right. \right. \\ & \left. \left. + e^z)^{1/12}) - \arctan(2(1 + e^z)^{1/12} - \sqrt{3}) - \arctan(2(1 + e^z)^{1/12} + \sqrt{3}) \right. \\ & \left. - 2\sqrt{3} \arctan \left(\frac{1}{3} (2(1 + e^z)^{1/12} + 1) \sqrt{3} \right) + \sqrt{3} \arctan \left(\frac{\sqrt{3}}{1 + 2(1 + e^z)^{1/6}} \right) \right. \\ & \left. + \frac{1}{2} \sqrt{3} \ln((1 + e^z)^{1/6} - \sqrt{3}(1 + e^z)^{1/12} + 1) - \frac{1}{2} \sqrt{3} \ln((1 + e^z)^{1/6} \right. \\ & \left. \left. + \sqrt{3}(1 + e^z)^{1/12} + 1) + 12(1 + e^z)^{1/12} \right) \right) \end{aligned}$$

with(CodeGeneration) :

$$\begin{aligned} & CSharp \left(\frac{h}{\text{Pi}} \cdot \text{simplify} \left(\text{subs} \left(x = (1 + \exp(z))^{\frac{1}{12}}, 12 \cdot \text{int} \left(\frac{x^{12}}{x^{12} - 1}, x \right) \right) - \text{subs} \left(z = I \cdot \text{Pi}, \text{subs} \left(\right. \right. \right. \right. \\ & \left. \left. \left. \left. x = (1 + \exp(z))^{\frac{1}{12}}, 13 \cdot \text{int} \left(\frac{x^{12}}{x^{12} - 1}, x \right) \right) \right) + I \cdot \text{Pi} \right) \right) \end{aligned}$$

Остання команда проводить генерацію коду на мові C# із використанням математичних методів класу System.Math, але тому, що в даній роботі був використаний інший клас, що був розроблений автором даної роботи, назви методів (функцій) було змінено. Також була змінена назва аргументу (z було змінено на dzeta).

Обчислення інтегралу

$$z = C \int_0^{\zeta} (\zeta - 1)^{\alpha-1} (\zeta + 1)^{\alpha-1} \zeta^{2-2\alpha} d\zeta,$$

а також знаходження невідомої константи C проводилось в системі Wolfram Mathematica 8 наступним чином:

```
c * Integrate[((t^2 - 1)^(a - 1)) * (t^(2 - 2*a)), {t, 0, z}]
ConditionalExpression[
c \left( z \text{Hypergeometric2F1}\left[-\frac{1}{2}, 1-a, \frac{1}{2}, \frac{1}{z^2}\right] + \frac{i \sqrt{\pi} \text{Gamma}[a] \text{Tan}[a \pi]}{\text{Gamma}\left[-\frac{1}{2}+a\right]}\right), 0 < \text{Re}[a] < \frac{3}{2} \&& \text{Re}[z] > 1 \&& \text{Im}[z] == 0]

c * Integrate[((t^2 - 1)^(a - 1)) * (t^(2 - 2*a)), {t, -1, 1}]
ConditionalExpression[-\frac{2 c \text{Cos}[a \pi] \text{Gamma}\left[\frac{3}{2}-a\right] \text{Gamma}[a]}{\sqrt{\pi}}, 0 < \text{Re}[a] < \frac{3}{2}]

Solve[-\frac{2 c \text{Cos}[a \pi] \text{Gamma}\left[\frac{3}{2}-a\right] \text{Gamma}[a]}{\sqrt{\pi}} == A, c]
{{c \rightarrow -\frac{A \sqrt{\pi} \text{Sec}[a \pi]}{2 \text{Gamma}\left[\frac{3}{2}-a\right] \text{Gamma}[a]}}}

{f} = {Function[{z}, -\frac{A \sqrt{\pi} \text{Sec}[a \pi]}{2 \text{Gamma}\left[\frac{3}{2}-a\right] \text{Gamma}[a]} \left( z \text{Hypergeometric2F1}\left[-\frac{1}{2}, 1-a, \frac{1}{2}, \frac{1}{z^2}\right] + \frac{i \sqrt{\pi} \text{Gamma}[a] \text{Tan}[a \pi]}{\text{Gamma}\left[-\frac{1}{2}+a\right]}\right)]};
```

Остаточний результат являє собою функцію, за допомогою якої, до речі, проводилася верифікація реалізованих чисельних методів для обчислення гамма-функції та гіпергеометричної функції.

2. Лістинг вихідного коду програми

2.1. Клас комплексного потенціала

```
/// <summary>
/// Клас, що інкапсулює функціонал комплексного потенціалу течії
/// Спадкує інтерфейс, що має подію, яка повідомляє про зміну
/// параметрів комплексного потенціалу
/// </summary>
class Potential : INotifyPropertyChanged
{
    /// <summary>
    /// Індексатор, що повертає значення комплексного потенціалу в точці
    /// </summary>
    /// <param name="dzeta">Комплексне число, що являє собою точку, в якій знаходимо значення
    /// потенціалу</param>
    public Complex this[Complex dzeta] { get { return W(dzeta); } }

    /// <summary>
    /// Швидкість потоку в нескінченності
    /// </summary>
    double _V_inf;

    /// <summary>
    /// Кут атаки
    /// </summary>
    double _alpha;

    /// <summary>
    /// Радіус циліндра, що обтікається
    /// </summary>
    double _R;

    /// <summary>
    /// Циркуляція
    /// </summary>
    double _G;

    /// <summary>
    /// Інтерфейсне посилання, що являє собою функцію конформного відображення.
    /// Функція являє собою тип даних, який реалізує інтерфейс IConformalMapFunction.
    /// Такий підхід дозволяє використовувати поліморфізм в коді програми, маючи один клас,
    /// що представляє собою комплексний потенціал, і поліморфне посилання, до якого будуть
    /// неявно зводитися будь-які типи, які реалізують потрібний інтерфейс
    /// </summary>
    IConformalMapFunction _f;
```

```

/// <summary>
/// Швидкість на нескінченності
/// </summary>
public double V_inf
{
    get
    {
        return _V_inf;
    }
    set
    {
        //Не може бути нульовою або від'ємною
        _V_inf = value == 0 ? throw new ArgumentException() : Abs(value);
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

/// <summary>
/// Радіус обтекаемого циліндра
/// </summary>
public double R
{
    get
    {
        return _R;
    }
    set
    {
        //Не може бути нульовим або від'ємним (якщо не задається в конструкторі)
        _R = value == 0 ? throw new ArgumentException() : Abs(value);
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

/// <summary>
/// Циркуляція потоку
/// </summary>
public double G
{
    get
    {
        return _G;
    }
    set
    {
        _G = value;
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

```

```

/// <summary>
/// Кут атаки в радіанах
/// </summary>
public double AlphaRadians
{
    get
    {
        return _alpha;
    }
    set
    {
        _alpha = value;
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

/// <summary>
/// Кут атаки в градусах
/// </summary>
public double AlphaDegrees
{
    get
    {
        return _alpha * 180.0 / Math.PI; ;
    }
    set
    {
        _alpha = value * Math.PI / 180.0;
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

/// <summary>
/// Інтерфейсне посилання, що являє собою функцію конформного відображення.
/// Функція являє собою тип даних, який реалізує інтерфейс IConformalMapFunction.
/// Такий підхід дозволяє використовувати поліморфізм в коді програми, маючи один клас,
/// що представляє собою комплексний потенціал, і поліморфне посилання, до якого будуть
/// неявно зводитися будь-які типи, які реалізують потрібний інтерфейс
/// </summary>
public IConformalMapFunction f
{
    get { return _f; }
    set
    {
        _f = value;
        //Виконання метода з подією повідомлення (не використовується
        //в даній реалізації)
        OnPropertyChanged();
    }
}

/// <summary>
/// Тимчасова змінна
/// </summary>
Complex tmp;

```

```

/// <summary>
/// Подія, яка виконується при зміні якогось параметра
/// </summary>
public event PropertyChangedEventHandler PropertyChanged;

/// <summary>
/// Конструктор екземпляра класу, що інкапсулює функціонал комплексного потенціалу течії
/// </summary>
/// <param name = "V_inf"> Швидкість течії у нескінченності </param>
/// <param name = "alpha"> Кут атаки в радіанах. Повинен дорівнювати нулю в разі обтікання
смуги або півплощини </param>
/// <param name = "R"> Радіус циліндра, що обтікається. Повинен дорівнювати нулю в разі
обтікання смуги або півплощини </param>
/// <param name = "G"> Циркуляція. Повинна дорівнювати нулю в разі обтікання смуги або
півплощини </param>
/// <param name = "f"> Функція конформного відображення (реалізує інтерфейс
IConformalMapFunction) </param>
public Potential(double V_inf, double alpha, double R, double G, IConformalMapFunction f)
{
    this.f = f;
    this._V_inf = V_inf;
    this._R = R;
    this._alpha = R == 0 ? 0 : alpha;
    this._G = R == 0 ? 0 : G;
}

/// <summary>
/// Потенціальна функція
/// </summary>
/// <param name="z">Точка, в якій знаходимо значення потенціальної функції</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public double phi(Complex z)
{
    return this[z].Re;
}

/// <summary>
/// Функція струму
/// </summary>
/// <param name="z">Точка, в якій знаходимо значення функції струму</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public double psi(Complex z)
{
    return this[z].Im;
}

/// <summary>
/// Швидкість у допоміжній площині у вигляді комплексного числа
/// </summary>
/// <param name="dzeta">Точка, в якій знаходимо значення швидкості</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Complex V(Complex dzeta)
{
    return dW_ddzeta(dzeta).Conjugate;
}

```

```

/// <summary>
/// Швидкість у фізичній площині у вигляді комплексного числа
/// </summary>
/// <param name="z">Точка, в якій знаходимо значення швидкості</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Complex V_physical_plane(Complex z)
{
    try
    {
        //Знаходимо значення оберненої функції конформного відображення в точці.
        //У випадку будь-якої помилки повертаємо НЕ-число (NaN).
        //В іншому випадку знаходимо швидкість (за правилом диференціювання
        //складної функції)
        tmp = f.dzeta(z);
        return IsNaN(tmp) ? NaN : (dW_ddzeta(z) / f.dz_ddzeta(tmp)).Conjugate;
    }
    catch { return NaN; }
}

/// <summary>
/// Горизонтальна компонента швидкості у допоміжній площині
/// </summary>
/// <param name="dzeta">Точка, в якій знаходимо значення горизонтальної компоненти
/// швидкості</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public double V_ksi(Complex dzeta)
{
    return dW_ddzeta(dzeta).Conjugate.Re;
}

/// <summary>
/// Вертикальна компонента швидкості у допоміжній площині
/// </summary>
/// <param name="dzeta">Точка, в якій знаходимо значення вертикальної компоненти
/// швидкості</param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public double V_eta(Complex dzeta)
{
    return dW_ddzeta(dzeta).Conjugate.Im;
}

/// <summary>
/// Значення комплексного потенціала в точці
/// </summary>
/// <param name="dzeta"> Точка, в якій знаходимо значення комплексного потенціала </param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Complex W(Complex dzeta)
{
    return _V_inf * Exp(-I * this._alpha) * dzeta + (_R * _R * _V_inf * Exp(I * _alpha)) /
dzeta + this._G * Ln(dzeta) / (2 * Math.PI * I);
}

```

```

/// <summary>
/// Значення похідної від функції комплексного потенціала в точці
/// </summary>
/// <param name="dzeta"> Точка, в якій знаходимо значення похідної </param>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Complex dW_ddzeta(Complex dzeta)
{
    //Комплексний потенціал заданий у загальному вигляді для
    return _V_inf * Exp(-I * this._alpha) - (_R * _R * _V_inf * Exp(I * this._alpha)) /
(dzeta * dzeta) + this._G / (2 * Math.PI * I * dzeta);
}

/// <summary>
/// Строкове представлення комплексного потенціала
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public override string ToString()
{
    return $"V={_V_inf}, alpha={_alpha}, R={_R}, G={_G}";
}

/// <summary>
/// Метод, що визивається при виконанні події PropertyChanged
/// </summary>
public void OnPropertyChanged([CallerMemberName] string propertyName = null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

2.2. Інтерфейс функції конформного відображення

Слід зазначити, що в даному розділі нема різниці між типом комплексного числа та точки, а в самому коді передбачена можливість приведення комплексного числа к типу точки та навпаки.

```

/// <summary>
/// Інтерфейс, що має в собі загальні методи і індексатор для функції конформного
/// відображення.
/// Використання інтерфейсного посилання дозволяє використовувати поліморфізм в коді,
/// визначивши
/// в класі комплексного потенціалу змінну типу інтерфейсу і використовувати її для безлічі
/// конформних відображень,
/// що визначені в програмі.
/// </summary>
interface IConformalMapFunction
{
    /// <summary>
    /// Індексатор з доступом тільки для читання, що приймає об'єкт точки і
    /// повертає значення функції конформного відображення (також у вигляді точки)
    /// </summary>
    DataPoint this[DataPoint dzeta] { get; }

    /// <summary>
    /// Значення функції в точці (у вигляді комплексного числа)
    /// </summary>
    Complex z(Complex dzeta);
}

```

```

/// <summary>
/// Значення похідної функції в точці (у вигляді комплексного числа)
/// </summary>
Complex dz_ddzeta(Complex dzeta);

/// <summary>
/// Значення функції, оберненої до функції конформного відображення
/// (у вигляді комплексного числа)
/// </summary>
Complex dzeta(Complex Z);

/// <summary>
/// Значення функції в точці (у вигляді точки)
/// </summary>
DataPoint z(DataPoint dzeta);

/// <summary>
/// Значення похідної функції в точці (у вигляді точки)
/// </summary>
DataPoint dz_ddzeta(DataPoint dzeta);

/// <summary>
/// Значення функції, оберненої до функції конформного відображення
/// (у вигляді точки)
/// </summary>
DataPoint dzeta(DataPoint Z);
}

```

2.3. Структура комплексного числа

```

/// <summary>
/// Структура, що являє собою комплексне число
/// </summary>
[Serializable]
public struct Complex : IEquatable<Complex>, IFormattable
{
    /// <summary>
    /// Дійсна частина комплексного числа
    /// </summary>
    public readonly double Re;

    /// <summary>
    /// Уявна частина комплексного числа
    /// </summary>
    public readonly double Im;

    /// <summary>
    /// Модуль комплексного числа
    /// </summary>
    public double Abs => Math.Sqrt(Re * Re + Im * Im);

    /// <summary>
    /// Аргумент комплексного числа (радіани)
    /// </summary>
    public double ArgRadians => getArg();

    /// <summary>
    /// Аргумент комплексного числа (градуси)
    /// </summary>
    public double ArgDegrees => getArg() * 180.0 / Math.PI;
}

```

```

/// <summary>
/// Спряжене комплексне число
/// </summary>
public Complex Conjugate => new Complex(Re, -Im);

/// <summary>
/// Операція спряження
/// </summary>
public static Complex ComplexConjugate(Complex z) => z.Conjugate;

/// <summary>
/// Мніма одиниця
/// </summary>
public static Complex I => new Complex(0, 1);

/// <summary>
/// Нуль
/// </summary>
public static Complex Zero => new Complex();

/// <summary>
/// Нескінченно віддалена точка
/// </summary>
public static Complex Infinity => new Complex(double.PositiveInfinity,
double.PositiveInfinity);

/// <summary>
/// НЕ-число
/// </summary>
public static Complex NaN => new Complex(double.NaN, double.NaN);

/// <summary>
/// Комплексне число, дійсна та уявна частини якого є дуже малими величинами
/// </summary>
public static Complex Epsilon => new Complex(double.Epsilon, double.Epsilon);

/// <summary>
/// Аргумент комплексного числа
/// </summary>
/// <param name="m">Спосіб вимірювання кута (градуси чи радіани)</param>
/// <returns></returns>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public double GetArgument(AngleMeasurement m)
{
    switch (m)
    {
        case AngleMeasurement.Degrees:
            return getArg() * 180.0 / Math.PI;
        case AngleMeasurement.Radians:
            return getArg();
        default:
            throw new ArgumentOutOfRangeException("Invalid angle measurement");
    }
}

```

```

[MethodImpl(MethodImplOptions.AggressiveInlining)]
private double getArg()
{
    if ((Re > 0 && Im > 0) || (Re > 0 && Im < 0))
    {
        return Math.Atan(Im / Re);
    }
    else if ((Re < 0 && Im > 0) || (Re < 0 && Im < 0))
    {
        return Math.PI + Math.Atan(Im / Re);
    }
    else if (Im == 0 && Re > 0)
    {
        return 0;
    }
    else if (Im == 0 && Re < 0)
    {
        return Math.PI;
    }
    else if (Re == 0 && Im < 0)
    {
        return 3 * Math.PI / 2;
    }
    else if (Re == 0 && Im > 0)
    {
        return Math.PI / 2;
    }
    else
    {
        return 0;
    }
}

/// <summary>
/// Конструктор комплексного числа
/// </summary>
public Complex(double Re, double Im)
{
    this.Re = Re;
    this.Im = Im;
}

/// <summary>
/// Перевірка на рівність
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public bool Equals(Complex other)
{
    return Re == other.Re && Im == other.Im;
}

/// <summary>
/// Отримання хеш-коду комплексного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public override int GetHashCode()
{
    return Re.GetHashCode() ^ Im.GetHashCode();
}

```

```

/// <summary>
/// Строкове представлення комплексного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public override string ToString()

/// <summary>
/// Комплексне число, отримане по значенню модуля та аргументу у радіанах
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex ComplexNumberFromMagnitudeAndPhaseInRadians(double r, double phiRad)

/// <summary>
/// Комплексне число, отримане по значенню модуля та аргументу у градусах
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex ComplexNumberFromMagnitudeAndPhaseInDegrees(double r, double phiDeg)

#region MathOperators

/// <summary>
/// Сума двох комплексних чисел
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator +(Complex c1, Complex c2)

/// <summary>
/// Сума комплексного та дійсного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator +(double d, Complex c)

/// <summary>
/// Сума комплексного та дійсного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator +(Complex c, double d)

/// <summary>
/// Унарний плюс
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator +(Complex c)

/// <summary>
/// Різниця двох комплексних чисел
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator -(Complex c1, Complex c2)

/// <summary>
/// Різниця дійсного та комплексного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator -(double d, Complex c)

```

```

/// <summary>
/// Різниця комплексного та дійсного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator -(Complex c, double d)

/// <summary>
/// Унарний мінус
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator -(Complex c)

/// <summary>
/// Добуток двох комплексних чисел
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator *(Complex c1, Complex c2)

/// <summary>
/// Добуток дійсного та комплексного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator *(double d, Complex c)

/// <summary>
/// Добуток комплексного та дійсного числа
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator *(Complex c, double d)

/// <summary>
/// Ділення одного комплексного числа на інше
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator /(Complex z1, Complex z2)

/// <summary>
/// Ділення дійсного числа на комплексне
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator /(double d, Complex z2)

/// <summary>
/// Ділення комплексного числа на дійсне
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator /(Complex z1, double d)

/// <summary>
/// Перевірка на рівність двох комплексних чисел
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static bool operator ==(Complex c1, Complex c2)

/// <summary>
/// Перевірка на нерівність двох комплексних чисел
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static bool operator !=(Complex c1, Complex c2)

```

```

/// <summary>
/// Декремент (діє тільки на дійну частину)
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator --(Complex z)

/// <summary>
/// Інкремент (діє тільки на дійсну частину)
/// </summary>
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static Complex operator ++(Complex z)

#endregion
}

```

2.4. Деякі окремі методи

```

/// <summary>
/// Розв'язок рівняння виду f(z)=Z. Початкові наближення є випадковими величинами
/// </summary>
public static Complex Solve(Func<Complex, Complex> f, Func<Complex, Complex> df, Complex Z)
{
    //Початкове наближення спочатку є правою частиною
    Complex.Complex initial_approximation = Z, z, tmp;
    //кількість невдач при розрахунках та кількість розрахунків
    int fails = 0, iters;
    Func<Complex.Complex, Complex.Complex> F = (c) => { return f(c) - Z; }, dFdz = (c) =>
{ return df(c); };
    //Екземпляр класу, призначений для роботи з псевдовипадковими величинами
    Random rnd = new Random(DateTime.Now.Millisecond);
    //Точка початку ітераційного процесу при невдачі
    fail_mark:
    //До початкового наближення додаємо псевдовипадкове комплексне число
    initial_approximation += (new Complex.Complex(rnd.NextDouble(), rnd.NextDouble()));
    z = initial_approximation;
    tmp = z;
    iters = 0;
    while (fails < 50)
    {
        iters++;
        //Процес метода Ньютона
        z = z - F(z) / dFdz(z);
        //Якщо досягнута точність
        if ((z - tmp).Abs <= Settings.Eps)
        {
            //Перевіряємо, чи буде знайдена точка, при дії на неї вказаної
            //функції конформного відображення правою частиною
            if ((f(z) - Z).Abs <= Settings.Eps) { return z; }
            //Якщо ні, то інкрементуємо кількість невдач та переходимо
            //в початок ітераційного процесу
            else { fails++; goto fail_mark; }
        }
        //Якщо результат НЕ-число, нескінченно віддалена точка або кількість розрахунків
        // стала більшою за 100, то інкрементуємо кількість невдач та переходимо
        // в початок ітераційного процесу
        if (Complex.Complex.IsNaN(z) || Complex.Complex.IsInfinity(z) || iters > 100)
        {
            fails++; goto fail_mark;
        }
        tmp = z;}
    //Якщо кількість невдач стала більшою за 49, то повертаємо НЕ-число
    return Complex.Complex.NaN;
}

```

```

/// <summary>
/// Гіпергеометрична функція Гауса
/// </summary>
public static Complex Hypergeometric2F1(double a, double b, double c, Complex z)
{
    //c!=0,-1,-2,...
    if (z == 0) { return 1; }
    if (z.Abs > 1)
    {
        if (Abs(((z).ArgRadians)) < Math.PI)
        {
            return (Gamma(c) * Gamma(b - a) / (Gamma(b) * Gamma(c - a))) *
Hypergeometric2F1(a, a + 1 - c, a + 1 - b, 1.0 / z) * Pow(-z, -a) + (Gamma(c) * Gamma(a - b) /
(Gamma(a) * Gamma(c - b))) * Hypergeometric2F1(b, b + 1 - c, b + 1 - a, 1.0 / z) * Pow(-z, -b);
        }
        else
        {
            return ((Gamma(c) * Gamma(b - a) / (Gamma(b) * Gamma(c - a))) *
Hypergeometric2F1(a, a + 1 - c, a + 1 - b, 1.0 / z.Conjugate) * Pow((-z).Conjugate, -a) +
(Gamma(c) * Gamma(a - b) / (Gamma(a) * Gamma(c - b))) * Hypergeometric2F1(b, b + 1 - c, b + 1 -
a, 1.0 / z.Conjugate) * Pow((-z).Conjugate, -b)).Conjugate;
        }
    }
    else
    {
        Complex sum = 1, tmp = sum, comp = 1;
        for (int k = 1; ; k++)
        {
            for (int l = 0; l <= k - 1; l++)
            {
                comp *= (a + l) * (b + l) / ((l + 1) * (c + l));
            }
            comp *= Pow(z, k);
            sum += comp;
            if ((sum - tmp).Abs <= Settings.Eps) { return sum; }
            tmp = sum;
            comp = 1;
        }
    }
}

```