

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра вычислительной математики

ДРОЗД Ольга Вячеславовна

**МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ ЧИСЛЕННОГО
ИССЛЕДОВАНИЯ РАВНОВЕСНЫХ ФОРМ КАПЛИ,
ВРАЩАЮЩЕЙСЯ В ПОЛЕ СИЛЫ ТЯЖЕСТИ**

Дипломная работа

Научный руководитель:
Ю.Н. Горбачёва

Допущена к защите

«_____» _____ 2025 г.

Заведующий кафедрой _____

кандидат физ.-мат. наук, доцент В.И. Репников

Минск, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
ГЛАВА 1 ПОСТАНОВКА ЗАДАЧИ	7
1.1 Вариационная постановка задачи	7
1.2 Метод множителей Лагранжа	13
ГЛАВА 2 МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ ДЛЯ НАХОЖДЕНИЯ РАВНОВЕСНЫХ ФОРМ КАПЛИ	17
2.1 Общие сведения	17
2.2 Схема конечных элементов	18
2.3 Метод Ньютона решения систем нелинейных уравнений	23
2.3.1 Общие сведения	23
2.3.2 Описание метода Ньютона	24
ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	27
3.1 Алгоритм нахождения равновесных форм капли.....	27
3.2 Поиск критических значений числа Вебера.....	28
3.3 Поиск экспериментального порядка точности	29
3.4 Полученные результаты	29
3.4.1 Графические результаты при различных определяющих параметрах	29
3.4.2 Критические значения числа Вебера	32
3.4.3 Экспериментальное определение порядка точности.....	32
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35
ПРИЛОЖЕНИЕ А.....	36

РЕФЕРАТ

Дипломная работа, 47 страниц, 6 рисунков, 3 таблицы, 9 источников.

Ключевые слова: КАПЛЯ, СВОБОДНАЯ ПОВЕРХНОСТЬ, ОСЕВАЯ СИММЕТРИЯ, ЧИСЛО БОНДА, ЧИСЛО ВЕБЕРА, УГОЛ СМАЧИВАНИЯ, МЕТОД НЬЮТОНА, МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ.

Объект исследования: капля жидкости, находящаяся на горизонтальной плоскости, вращающейся с постоянной угловой скоростью в поле силы тяжести.

Предмет исследования: построение равновесных форм вращающейся капли с помощью метода конечных элементов.

Цель исследования: численно исследовать равновесные формы и устойчивость капли, вращающейся на горизонтальной плоскости в поле силы тяжести, с использованием метода конечных элементов.

Метод исследования: метод конечных элементов.

Полученные результаты: рассмотрена вариационная постановка задачи, построен алгоритм метода конечных элементов для нахождения равновесных форм свободной поверхности капли, написана программная реализация вычислительного алгоритма на языке Java, найдены равновесные формы свободной поверхности капли при различных значениях определяющих параметров, найдены критические значения вращательного числа Вебера в зависимости от угла смачивания и числа Бонда, экспериментально исследован порядок точности метода.

Достоверность материалов и результатов дипломной работы: полученные в работе результаты согласуются с имеющимися литературными данными.

Область возможного практического применения: результаты могут использоваться для нанесения различных покрытий на твердых поверхностях центробежными способами, при производстве волокон и порошков.

РЭФЕРАТ

Дыпломная праца, 47 старонак, 6 малюнкаў, 3 табліцы, 9 крыніц.

Ключавыя словы: КРАПЛЯ, СВАБОДНАЯ ПАВЕРХНЯ, ВОСЕВАЯ СІМЭТРЫЯ, ЛІК БОНДА, ЛІК ВЭБЕРА, КУТ ЗМАЧВАННЯ, МЕТАД НЬЮТАНА, МЕТАД КАНЧАТКОВЫХ ЭЛЕМЕНТАЎ.

Аб'ект даследавання: кропля вадкасці, змешчаная на гарызантальнай плоскасці, якая верціцца з сталай кутняй хуткасцю ў поле сілы цяжару.

Прадмет даследавання: пабудова раўнаважкіх формаў якая верціцца кроплі з дапамогай метаду канчатковых элементаў.

Мэта даследавання: колькасна даследаваць раўнаважкія формы і ўстойлівасць кроплі, якая верціцца на гарызантальнай плоскасці ў полі сілы цяжару, з выкарыстаннем метаду канчатковых элементаў.

Метад даследавання: метад канчатковых элементаў.

Атрыманыя вынікі: разгледжана варыяцыйная пастаноўка задачы, пабудаваны алгарытм метаду канчатковых элементаў для знаходжання раўнаважных формаў свабоднай паверхні кроплі, напісана праграмная рэалізацыя вылічальнага алгарытму на мове Java, знойдзены раўнаважныя формы свабоднай паверхні кроплі пры розных значэннях вызначальных параметраў, знойдзены крытычныя значэнні круцільнага ліку Вэбера ў залежнасці ад кута змочвання і лікі Бонда, эксперыментальна даследаваны парадак дакладнасці метаду.

Дакладнасць матэрыялаў і вынікаў дыпломнай працы: атрыманыя ў працы вынікі адпавядаюць з навуковымі літаратурнымі дадзенымі.

Вобласць магчымага практычнага прымянення: вынікі могуць выкарыстоўвацца для нанясення розных пакрыццяў на цвёрдых паверхнях цэнтрабежнымі спосабамі, пры вытворчасці валокнаў і парашкоў.

REPORT

Diploma work, 47 pages, 6 drawings, 3 tables, 9 sources.

Keywords: DROP, FREE SURFACE, AXIAL SYMMETRY, BOND NUMBER, WEBER NUMBER, CONTACT ANGLE, NEWTON'S METHOD, FINITE ELEMENT METHOD.

The object of the research: a drop of liquid located on a horizontal plane rotating at a constant angular velocity in a gravity field.

The subject of the research: construction of equilibrium forms of a rotating drop using the finite element method

The purpose of the research: numerically investigate the equilibrium shapes and stability of a drop rotating on a horizontal plane in a gravity field using the finite element method.

Methods of research: finite element method.

The results of the work: a variational formulation of the problem is considered, an algorithm of the finite element method is constructed to find the equilibrium shapes of the free surface of a drop, a software implementation of the computational algorithm is written in Java, the equilibrium shapes of the free surface of a drop are found for various values of the defining parameters, critical values of the rotational Weber number are found depending on the wetting angle and the Bond number, the accuracy order of the method is experimentally investigated..

Authenticity of the materials and results of the diploma work: the results obtained in the work are consistent with the available literature data.

Recommendations on the usage: the results can be used for applying various coatings on hard surfaces by centrifugal methods, in the production of fibers and powders.

ВВЕДЕНИЕ

Капиллярные явления – не самый простой объект для исследования, но имеющий значимые теоретическое и практическое значения. Нелинейные дифференциальные уравнения и их системы являются математической основой для моделирования капиллярных поверхностей. Рассчитанные модели нашли свое применение в различных отраслях, включая металлургию, нефтяную промышленность, микроэлектронику и другие отрасли.

Для решения задачи о равновесных формах капиллярной жидкости в общем случае применяются исключительно численные методы. Если говорить, например, о задачах гидростатики, то в данном случае «все сводится к минимизации функционала, описывающего полную энергию системы газ – жидкость – твердое тело, либо же сводится к решению нелинейных дифференциальных уравнений Юнга-Лапласа, дополняемых интегральным условием сохранения объема жидкости» [2]. Если же задача решается вторым методом, то численное решение сводится к решению краевой задачи [6] или начальной задачи методом стрельбы [1], [5].

В первой главе данной работы рассмотрена вариационная постановка задачи. Задача условной минимизации сводится к задаче на безусловный экстремум с помощью метода Лагранжа. Во второй главе рассмотрен метод конечных элементов. В результате решение поставленной задачи сводится к системе нелинейных алгебраических уравнений методом Ньютона. И в третьей главе описаны алгоритм построения равновесных форм капли при заданных значениях чисел Бонда Bo и Вебера P , и угла смачивания α , а также алгоритм поиска критических значений числа Вебера P . Были получены численные результаты с помощью разработанных алгоритмов и экспериментально исследован порядок точности метода.

ГЛАВА 1

ПОСТАНОВКА ЗАДАЧИ

1.1 Вариационная постановка задачи

Рассмотрим каплю жидкости объема V , находящуюся в равновесии на горизонтальной плоскости под действием сил поверхностного натяжения и центробежных сил, когда плоскость и капля вращаются вокруг оси симметрии с постоянной угловой скоростью ω .

Уравнение свободной поверхности капли, вращающейся на горизонтальной плоскости, выводится из минимизации полной энергии капли $\tilde{E}$, которая состоит из энергии поверхностного натяжения, потенциальной энергии в гравитационном поле, потенциальной энергии в поле центробежных сил и энергии смачивания [2,5,8]:

$$\tilde{E} = \sigma |\tilde{\Gamma}| + \rho g \int_{\tilde{\Omega}} Z d\tilde{\Omega} - \frac{\omega^2 \rho}{2} \int_{\tilde{\Omega}} (X^2 + Y^2) d\tilde{\Omega} - \sigma \cos \alpha |\tilde{\Sigma}|, \quad (1.1)$$

где σ – коэффициент поверхностного натяжения;

$\tilde{\Gamma}$ – свободная поверхность капли (т.е. граница между жидкостью и воздухом);

$|\cdot|$ – площадь соответствующей поверхности;

ρ – плотность жидкости;

g – ускорение свободного падения;

$\tilde{\Omega}$ – объем, заполненный жидкостью, т.е. $|\tilde{\Omega}| = V$;

α – угол смачивания;

$\tilde{\Sigma}$ – смоченная область (т.е. граница между жидкостью и горизонтальной плоскостью).

Уравнение свободной поверхности следует из условия минимума полной энергии (1.1) при дополнительном условии, что объем капли

$$V = \int_{\tilde{\Omega}} 1 d\tilde{\Omega} \quad (1.2)$$

задан.

Выполним замену переменных:

$$E = \frac{\tilde{E}}{\sigma V^{\frac{2}{3}}}, \quad x = \frac{X}{V^{\frac{1}{3}}}, \quad y = \frac{Y}{V^{\frac{1}{3}}}, \quad z = \frac{Z}{V^{\frac{1}{3}}}, \quad (1.3)$$

где E – безразмерная полная энергия капли.

Используя безразмерные переменные (1.3) перепишем формулу полной энергии (1.1):

$$\begin{aligned} E = \frac{\tilde{E}}{\sigma V^{\frac{2}{3}}} &= \frac{\sigma}{\sigma V^{\frac{2}{3}}} \left(|\tilde{\Gamma}| + \frac{\rho g}{\sigma} \int_{\tilde{\Omega}} Z d\tilde{\Omega} - \frac{\omega^2 \rho}{2\sigma} \int_{\tilde{\Omega}} (X^2 + Y^2) d\tilde{\Omega} - \cos \alpha |\tilde{\Sigma}| \right) \\ &= \\ &= \frac{1}{V^{\frac{2}{3}}} \left(V^{\frac{2}{3}} \int_{\Gamma} 1 d\Gamma + \frac{\rho g}{\sigma} \int_{\Omega} z V^{\frac{1}{3}} V d\Omega - \frac{\omega^2 \rho}{2\sigma} \int_{\Omega} (x^2 + y^2) V^{\frac{2}{3}} V d\Omega - \right. \\ &\quad \left. - \cos \alpha \int_{\Sigma} V^{\frac{2}{3}} 1 d\Sigma \right) = \int_{\Gamma} 1 d\Gamma + \frac{\rho g}{\sigma} V^{\frac{2}{3}} \int_{\Omega} z d\Omega - \\ &\quad - \frac{\omega^2 \rho}{2\sigma} V \int_{\Omega} (x^2 + y^2) d\Omega - \cos \alpha \int_{\Sigma} 1 d\Sigma = \int_{\Gamma} 1 d\Gamma + Bo \int_{\Omega} z d\Omega - \\ &\quad - P \int_{\Omega} (x^2 + y^2) d\Omega - \cos \alpha \int_{\Sigma} 1 d\Sigma, \end{aligned} \quad (1.4)$$

где Γ, Ω и Σ – безразмерная свободная поверхность, заполненная жидкостью область и смоченная область соответственно;

$Bo = \frac{\rho g}{\sigma} V^{\frac{2}{3}}$ – число Бонда, равное отношению гравитационных сил к капиллярным силам;

$P = \frac{\omega^2 \rho}{2\sigma} V$ – вращательное число Вебера, равное отношению между центробежными и капиллярными силами.

Так как

$$d\tilde{\Omega} = dX dY dZ = V^{\frac{1}{3}} dx V^{\frac{1}{3}} dy V^{\frac{1}{3}} dz = V d\Omega,$$

то условие (1.2) в переменных (1.3) примет вид

$$\int_{\Omega} 1 d\Omega = 1. \quad (1.5)$$

Опишем искомую осесимметричную форму капли Γ в виде [9]:

$$(x, y, z) = (u(\theta) \cos \phi \sin \theta, u(\theta) \sin \phi \sin \theta, u(\theta) \cos \theta), \quad (1.6)$$

$$\theta \in \left[0, \frac{\pi}{2}\right], \phi \in [0, 2\pi],$$

где $u(\theta)$ – расстояние от точки (x, y, z) на свободной поверхности Γ до начала координат;

θ – угол между осью симметрии Oz и отрезком, который соединяет начало координат с точкой (x, y, z) .

На рисунке 1.1 показаны осевое сечение капли и система координат.

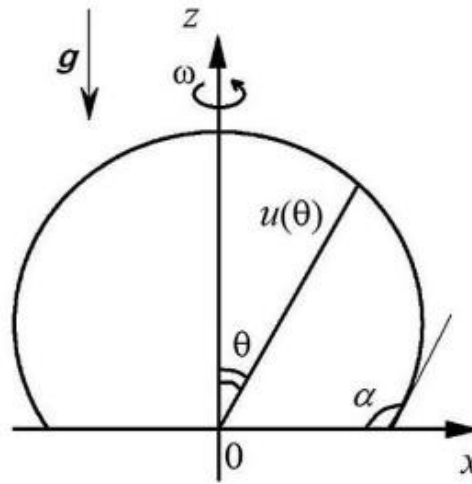


Рисунок 1.1 – Геометрические аспекты задачи

Выразим каждое слагаемое энергетического функционала (1.4) через сферические переменные (1.6).

Формула поверхностной энергии в новых переменных примет вид:

$$\int_{\Gamma} 1 \, d\Gamma = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \sqrt{(X_\phi, X_\phi)^2 (X_\theta, X_\theta)^2 - (X_\phi, X_\theta)^2} \, d\phi \, d\theta,$$

где

$$X(\varphi) = \begin{pmatrix} -u(\theta) \sin \varphi \sin \theta \\ u(\theta) \cos \varphi \sin \theta \\ 0 \end{pmatrix},$$

$$X(\theta) = \begin{pmatrix} u'(\theta) \cos \varphi \sin \theta + u(\theta) \cos \varphi \cos \theta \\ u'(\theta) \sin \varphi \sin \theta + u(\theta) \sin \varphi \cos \theta \\ u'(\theta) \cos \theta + u(\theta) (-\sin \theta) \end{pmatrix},$$

$$u' = u'(\theta) = \frac{du}{d\theta}.$$

$$(X_\theta, X_\phi)^2 = -uu' \cos \phi \sin \phi \sin^2 \theta - u^2 \sin \phi \cos \phi \sin \theta \cos \theta + \\ + uu' \sin \phi \cos \phi \sin^2 \theta + u^2 \sin \phi \cos \phi \sin \theta \cos \theta = 0;$$

$$(X_\theta, X_\theta)^2 = (u' \cos \phi \sin \theta + u \cos \phi \cos \theta)^2 + (u' \sin \phi \sin \theta + u \cos \theta \sin \phi)^2 \\ + (u' \cos \theta - u \sin \theta)^2 = u'^2 + u^2;$$

$$(X_\phi, X_\phi)^2 = u^2 \sin^2 \phi \sin^2 \theta + u^2 \cos^2 \phi \sin^2 \theta = u^2 \sin^2 \theta.$$

В результате данное слагаемое приобретет вид:

$$\int_{\Gamma} 1 d\Gamma = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \sqrt{u^2 \sin^2 \theta (u'^2 + u^2)} d\phi d\theta = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} u \sqrt{u'^2 + u^2} \sin \theta d\phi d\theta = \\ = 2\pi \int_0^{\frac{\pi}{2}} u \sqrt{u'^2 + u^2} \sin \theta d\theta.$$

Для выражения второго и третьего слагаемых нам понадобится Якобиан преобразования

$$J = r^2 \sin \theta.$$

И теперь, подставляя это значение, а также значения x, y, z в сферических координатах (1.4) в (1.3) получим следующий вид интегралов:

$$\int_{\Omega} z d\Omega = \int_{\Omega} z dx dy dz = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} r \cos \theta J d\phi d\theta dr =$$

$$\begin{aligned}
&= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} r \cos \theta r^2 \sin \theta d\phi d\theta dr = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} r^3 \cos \theta \sin \theta d\phi d\theta dr = \\
&= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \frac{u^4}{4} \cos \theta \sin \theta d\phi d\theta = \frac{\pi}{2} \int_0^{\frac{\pi}{2}} u^4 \cos \theta \sin \theta d\theta ;
\end{aligned}$$

$$\int_{\Omega} (x^2 + y^2) d\Omega = \int_{\Omega} (x^2 + y^2) dx dy dz =$$

$$= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} (r^2 \cos^2 \phi \sin^2 \theta + r^2 \sin^2 \phi \sin^2 \theta) J d\phi d\theta dr =$$

$$= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} (r^2 \cos^2 \phi \sin^2 \theta + r^2 \sin^2 \phi \sin^2 \theta) r^2 \sin \theta d\phi d\theta dr =$$

$$= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \int_0^{u(\theta)} r^4 \sin^3 \theta d\phi d\theta dr = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \frac{u^5}{5} \sin^3 \theta d\phi d\theta = \frac{2\pi}{5} \int_0^{\frac{\pi}{2}} u^5 \sin^3 \theta d\theta .$$

Преобразуем четвертое слагаемое:

$$\int_{\Sigma} 1 d\Sigma = \int_0^{2\pi} \int_0^{u(\frac{\pi}{2})} \sqrt{(X_{\phi}, X_{\phi})^2 (X_u, X_u)^2 - (X_{\phi}, X_u)^2} dr d\phi,$$

где

$$X_u = \begin{pmatrix} \cos \varphi \sin \theta \\ \sin \varphi \sin \theta \\ \cos \theta \end{pmatrix}.$$

Тогда

$$(X_u, X_u)^2 = \cos^2 \phi \sin^2 \theta + \sin^2 \phi \sin^2 \theta + \cos^2 \theta = 1;$$

$$(X_u, X_\phi)^2 = -u \sin \phi \sin \theta \cos \phi \sin \theta + u \cos \phi \sin \theta \sin \phi \sin \theta = 0.$$

В итоге:

$$\begin{aligned} \int_{\Sigma} 1 \, d\Sigma &= \int_0^{2\pi} \int_0^{u(\frac{\pi}{2})} \sqrt{(X_\phi, X_\phi)^2 (X_u, X_u)^2 - (X_\phi, X_u)^2} \, dr \, d\phi = \\ &= \int_0^{2\pi} \int_0^{u(\frac{\pi}{2})} u \sin \theta \, dr \, d\phi = \frac{u^2(\frac{\pi}{2})}{2} \cdot \sin \frac{\pi}{2} \cdot 2\pi = \pi \cdot u^2 \left(\frac{\pi}{2} \right). \end{aligned}$$

В результате энергетический функционал (1.4) в переменных (1.6) примет вид:

$$\begin{aligned} E(u) &= 2\pi \int_0^{\frac{\pi}{2}} u \sqrt{u'^2 + u^2} \sin \theta \, d\theta + \frac{\pi}{2} B o \int_0^{\frac{\pi}{2}} u^4 \cos \theta \sin \theta \, d\theta - \\ &\quad - \frac{2\pi}{5} P \int_0^{\frac{\pi}{2}} u^5 \sin^3 \theta \, d\theta - u^2 \pi \cos \alpha|_{\theta=\frac{\pi}{2}}, \end{aligned} \quad (1.7)$$

а условие (1.5) в переменных (1.6) примет вид:

$$\begin{aligned} \int_{\Omega} 1 \, d\Omega &= \int_{\Omega} 1 \, dx dy dz = \int_0^{2\pi} \int_0^{\frac{\pi}{2} u(\theta)} \int_0^{\frac{\pi}{2} u(\theta)} J \, d\phi \, d\theta \, dr = \\ &= 2\pi \int_0^{\frac{\pi}{2} u(\theta)} \int_0^{\frac{\pi}{2} u(\theta)} r^2 \sin \theta \, d\theta \, dr = 2\pi \int_0^{\frac{\pi}{2}} \frac{r^3}{3} \sin \theta|_0^{u(\theta)} \, d\theta = \\ &= \frac{2\pi}{3} \int_0^{\frac{\pi}{2}} u^3 \sin \theta \, d\theta. \end{aligned} \quad (1.8)$$

1.2 Метод множителей Лагранжа

Условие сохранения объема (1.8) включим в энергетический функционал (1.7) на основе метода множителей Лагранжа. Тем самым задачу условной минимизации (1.7), (1.8) сведем к задаче на безусловный экстремум функционала с неопределенным множителем Лагранжа λ [9]:

$$E_\lambda = E(u) + \lambda \left(\frac{2\pi}{3} \int_0^{\frac{\pi}{2}} u^3 \sin \theta \, d\theta - 1 \right). \quad (1.9)$$

В итоге получим следующий вид энергетического функционала (1.9) в переменных (1.6):

$$\begin{aligned} E_\lambda(u, \lambda) = & 2\pi \int_0^{\frac{\pi}{2}} u \sqrt{(u')^2 + u^2} \sin \theta \, d\theta + \\ & + \frac{\pi}{2} B o \int_0^{\frac{\pi}{2}} u^4 \sin \theta \cos \theta \, d\theta - \frac{2\pi}{5} P \int_0^{\frac{\pi}{2}} u^5 \sin^3 \theta \, d\theta - \\ & - \pi \cos \alpha u^2|_{\theta=\frac{\pi}{2}} + \lambda \left(\frac{2\pi}{3} \int_0^{\frac{\pi}{2}} u^3 \sin \theta \, d\theta - 1 \right). \end{aligned} \quad (1.10)$$

Теперь необходимо найти функцию $u: [0, \frac{\pi}{2}] \rightarrow \mathbb{R}$, которая минимизирует функционал (1.10). Необходимым условием минимума энергетического функционала является равенство нулю первой вариации функционала. Посчитаем эту вариацию, расписывая каждое слагаемое по отдельности и по окончании расчетов запишем единую формулу.

Первое слагаемое вариации:

$$2\pi \frac{d}{dt} \left[\int_0^{\frac{\pi}{2}} (u + tv) \sqrt{(u + tv)'^2 + (u + tv)^2} \sin \theta \, d\theta \right]_{t=0} =$$

$$\begin{aligned}
&= 2\pi \int_0^{\frac{\pi}{2}} (v\sqrt{(u' + tv')^2 + (u + tv)^2} + \\
&+ \frac{1}{2}(u + tv)((u' + tv')^2 + (u + tv)^2)^{-\frac{1}{2}} \times \\
&\times (2(u' + tv)v' + 2(u + tv)v)) \sin \theta d\theta |_{t=0} = \\
&= 2\pi \int_0^{\frac{\pi}{2}} \left(v\sqrt{u'^2 + u^2} + \frac{1}{2} \frac{u \cdot 2(u'v' + uv)}{\sqrt{u'^2 + u^2}} \right) \sin \theta d\theta = \\
&= 2\pi \int_0^{\frac{\pi}{2}} \frac{v(u'^2 + u^2) + u(u'v' + uv)}{\sqrt{u'^2 + u^2}} \sin \theta d\theta = \\
&= 2\pi \int_0^{\frac{\pi}{2}} \frac{v(u'^2 + 2u^2) + uu'v'}{\sqrt{u'^2 + u^2}} \sin \theta d\theta.
\end{aligned}$$

Второе слагаемое:

$$\begin{aligned}
&\frac{\pi}{2} Bo \frac{d}{dt} \left[\int_0^{\frac{\pi}{2}} (u + tv)^4 \sin \theta \cos \theta d\theta \right] |_{t=0} = \\
&= \frac{\pi}{2} Bo \int_0^{\frac{\pi}{2}} 4v(u + tv)^3 \sin \theta \cos \theta d\theta |_{t=0} = 2\pi Bo \int_0^{\frac{\pi}{2}} vu^3 \sin \theta \cos \theta d\theta.
\end{aligned}$$

Третье слагаемое:

$$-\frac{2\pi}{5} P \frac{d}{dt} \left[\int_0^{\frac{\pi}{2}} (u + tv)^5 \sin^3 \theta d\theta \right] |_{t=0} =$$

$$\begin{aligned}
&= -\frac{2\pi}{5} P \int_0^{\frac{\pi}{2}} 5 (u + tv)^4 v \sin^3 \theta \, d\theta|_{t=0} = \\
&= -\frac{2\pi}{5} P \int_0^{\frac{\pi}{2}} 5 u^4 v \sin^3 \theta \, d\theta = -2\pi P \int_0^{\frac{\pi}{2}} u^4 v \sin^3 \theta \, d\theta.
\end{aligned}$$

Четвертое слагаемое:

$$-\pi \cos \alpha \frac{d}{dt} [(u + tv)^2]|_{t=0} = -\pi \cos \alpha \cdot 2(u + tv)v|_{t=0} = -2\pi v u \cos \alpha.$$

Пятое слагаемое:

$$\begin{aligned}
&\frac{2\pi}{3} \frac{d}{dt} \left[\int_0^{\frac{\pi}{2}} (\lambda + t\mu)(u + tv)^3 \sin \theta \, d\theta \right] |_{t=0} = \\
&= \frac{2\pi}{3} \int_0^{\frac{\pi}{2}} (\mu(u + tv)^3 + (\lambda + t\mu) \cdot 3(u + tv)^2 v) \sin \theta \, d\theta = \\
&= \frac{2\pi}{3} \int_0^{\frac{\pi}{2}} (\mu u^3 + 3\lambda u^2 v) \sin \theta \, d\theta = \frac{2\pi}{3} \int_0^{\frac{\pi}{2}} \mu u^3 \sin \theta \, d\theta + 2\pi \int_0^{\frac{\pi}{2}} \lambda u^2 v \sin \theta \, d\theta.
\end{aligned}$$

И последнее слагаемое:

$$\frac{d}{dt} [-(\lambda + t\mu)]|_{t=0} = -\mu.$$

Теперь соберем все воедино и получим:

$$\begin{aligned}
\frac{d}{dt} E_\lambda(u + tv, \lambda + t\mu)|_{t=0} &= 2\pi \int_0^{\frac{\pi}{2}} \left(\frac{v(u'^2 + 2u^2) + uu'v'}{\sqrt{u'^2 + u^2}} + \right. \\
&\quad \left. + B o n u^3 \cos \theta - P u^4 v \sin^2 \theta + \lambda u^2 v + \frac{\mu}{3} u^3 \right) \sin \theta \, d\theta -
\end{aligned} \tag{1.11}$$

$$- 2\pi u v \cos \alpha |_{\theta=\frac{\pi}{2}} - \mu ,$$

где $v = v(\theta)$ – произвольная непрерывно дифференцируемая функция;
 t – параметр;
 μ – константа.

Если (1.11) сократить 2π и приравнять к нулю, а после выполнить подстановку $\mu = 0$, то для всех $v \in W$ получим следующее равенство:

$$\int_0^{\frac{\pi}{2}} \left(\frac{v(u'^2 + 2u^2) + uu'v'}{\sqrt{u'^2 + u^2}} + B_0 v u^3 \cos \theta - P u^4 v \sin^2 \theta + \right. \quad (1.12)$$

$$\left. + \lambda u^2 v \right) \sin \theta \, d\theta - uv \cos \alpha |_{\theta=\frac{\pi}{2}} = 0.$$

Равенство (1.12) «представляет собой уравнение равновесия жидкости, иначе еще называемое слабой формой нелинейного дифференциального уравнения Юнга-Лапласа с множителем Лагранжа. Также важно подчеркнуть, что в данном уравнении учтено граничное условие с заданным углом смачивания α » [2].

Если в формуле (1.11) выполнить подстановки $v = 0$ и $\mu = 1$, а также приравнять к нулю, то условие сохранения объема примет следующий вид

$$\frac{2\pi}{3} \int_0^{\frac{\pi}{2}} u^3 \sin \theta \, d\theta = 1. \quad (1.13)$$

После сделанного можно сделать вывод, что уравнение (1.12) и условие (1.13) описывают полную дифференциальную модель равновесия жидкости, а определяющими параметрами этой модели являются числа Бонда B_0 , Вебера P и угол смачивания α .

ГЛАВА 2

МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ ДЛЯ НАХОЖДЕНИЯ РАВНОВЕСНЫХ ФОРМ КАПЛИ

2.1 Общие сведения

Сам факт появления метода конечных элементов (МКЭ) связан с тем, что было необходимо найти новые варианты для решения задач строительной механики и теории упругости в 1930-х годах. У математиков есть для этого метода специфическое название – вариационно-разностный. Именно это название указывает на математическую природу подхода.

Первое пиковое достижение МКЭ относится к 1963 году, так как именно тогда появилось доказательство того, что такой метод можно рассматривать как один из вариантов известного в строительной механике метода Рэлея-Ритца, ведь он при переходе к минимизации потенциальной энергии позволяет свести задачу к системе линейных уравнений равновесия.

В момент окончательно принятого решения о том, что у МКЭ есть связь с минимизацией, появилась возможность использовать этот метод для решения задач в других областях техники. С помощью данного метода можно было решить задачи, описываемые уравнениями Лапласа или Пуассона, ибо решение этих уравнений также связано с минимизацией некоторого функционала. На данный момент есть информация о результатах решения задач распространения тепла, гидромеханики, течения жидкости в пористой среде и т.д. с применением метода.

Область применения МКЭ значительно увеличилась к концу 60-х, когда исследователи установили, что уравнения для элементов в задачах строительной механики, распространения тепла, гидромеханики, могут быть выведены через вариации метода взвешенных невязок, включая метод Галёркина или способ наименьших квадратов. Данное открытие стало ключевым в процессе теоретического обоснования МКЭ, т. к. увеличило его использование при решении многих типов дифференциальных уравнений [4].

Главный посыл метода конечных элементов состоит в том, что любую непрерывную величину можно аппроксимировать моделью, состоящей из отдельных элементов (участков). На каждом из этих элементов исследуемая непрерывная величина аппроксимируется кусочно-непрерывной функцией, которая строится на значениях исследуемой непрерывной величины в конечном числе точек рассматриваемого элемента.

2.2 Схема конечных элементов

Решение задачи (1.12) - (1.13) в данном случае будем искать, используя метод конечных элементов [4, 7] на равномерной сетке

$$\left\{ \theta_i = ih \mid i = \overline{0, N}; h = \frac{\pi}{2N} \right\}$$

Пусть $I_i = (\theta_{i-1}, \theta_i)$ – сеточный элемент, P_1 – пространство многочленов степени 0 либо 1.

Пусть имеется конечномерное подпространство

$$W_h = \left\{ v_h \in H^1 \left(0, \frac{\pi}{2} \right) : v_h|_{I_i} \in P_1 \right\}.$$

Приближенное решение задачи (1.12) - (1.13) найдем в виде

$$u_h(\theta) = \sum_{i=0}^N u_i \Phi_i(\theta), \quad (2.1)$$

где $\Phi_i(\theta) \in W_h, i = \overline{0, N} : \Phi_i(\theta_j) = \delta_{ij}, i, j = \overline{0, N}$ – финитные функции.

На рисунке 2.1 показано изображение финитных функций, которые еще называют «пирамидалными» или кусочно-линейными базисными функциями.

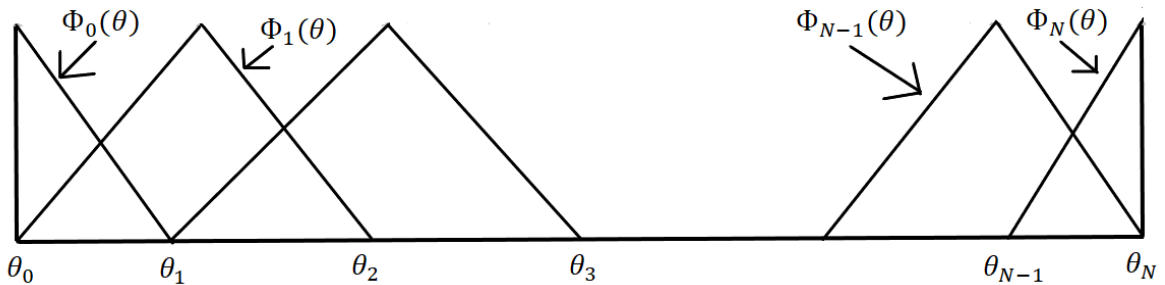


Рисунок 2.1 – Построение кусочно-линейных базисных функций

Очевидно их формульное представление:

$$\Phi_i(\theta) = \begin{cases} 0, & \theta \notin [\theta_{i-1}, \theta_{i+1}], \\ \frac{\theta_{i+1} - \theta}{h}, & \theta \in [\theta_i, \theta_{i+1}], \\ \frac{\theta - \theta_{i-1}}{h}, & \theta \in [\theta_{i-1}, \theta_i], i = \overline{1, N-1}; \end{cases}$$

$$\Phi_i(\theta) = \begin{cases} 0, & \theta \notin [\theta_0, \theta_1], \\ \frac{\theta_1 - \theta}{h}, & \theta \in [\theta_0, \theta_1]; \end{cases} \quad (2.2)$$

$$\Phi_i(\theta) = \begin{cases} 0, & \theta \notin [\theta_{N-1}, \theta_N], \\ \frac{\theta - \theta_{N-1}}{h}, & \theta \in [\theta_{N-1}, \theta_N]; \end{cases}$$

Интегралы, присутствующие в формулах (1.12) - (1.13), разбиваем на сумму интегралов и аппроксимируем квадратурной формулой трапеции, т.е.:

$$\int_0^{\frac{\pi}{2}} G(\theta) d\theta = \sum_{i=1}^N \int_{\theta_{i-1}}^{\theta_i} G(\theta) d\theta \approx \frac{h}{2} \sum_{i=1}^N (G(\theta_{i-1}) + G(\theta_i)). \quad (2.3)$$

Если положить, что $v_i = \Phi_i$, $i = \overline{0, N}$, то придем к системе нелинейных алгебраических уравнений. Для упрощения расчетов заменим все производные разностными производными.

В случае, когда производная вычисляется в точке $G(\theta_i)$, применяется левая разностная производная. В результате получим следующее:

$$u'(\theta_i) = \frac{u_i - u_{i-1}}{h},$$

$$\Phi_j'(\theta_i) = \frac{\Phi_j(\theta_i) - \Phi_j(\theta_{i-1})}{h}.$$

Сразу же можно заметить, с учетом ранее введенного символа Кронекера, что если

- $j = i$, то $\Phi_j'(\theta_i) = \frac{1-0}{h} = \frac{1}{h}$;

- $j = i - 1$, то $\Phi'_j(\theta_i) = \frac{0-1}{h} = -\frac{1}{h}$.

Если же производная находится в $G(\theta_{i-1})$, то будем пользоваться правой разностной производной. Тогда получим:

$$u'(\theta_{i-1}) = \frac{u_i - u_{i-1}}{h},$$

$$\Phi'_j(\theta_{i-1}) = \frac{\Phi_j(\theta_i) - \Phi_j(\theta_{i-1})}{h}.$$

Аналогично случаю выше, если

- $j = i$, то $\Phi'_j(\theta_{i-1}) = \frac{1-0}{h} = \frac{1}{h}$;
- $j = i - 1$, то $\Phi'_j(\theta_{i-1}) = \frac{0-1}{h} = -\frac{1}{h}$.

С учетом полученных равенств и подставляя в задачу (1.10) $v_i = \Phi_i$, $i = \overline{1, N}$ подсчитаем функции F.

- $i = 0$

$$\begin{aligned} G &= \int_{\theta_0}^{\theta_1} G(\theta) d\theta \approx \frac{h}{2} (G(\theta_0) + G(\theta_1)) = \frac{h}{2} (G(0) + G(\theta_1)) = \\ &= \frac{h}{2} \cdot \frac{u_1 \frac{u_1 - u_0}{h} \cdot \left(-\frac{1}{h}\right)}{\sqrt{\left(\frac{u_1 - u_0}{h}\right)^2 + u_1^2}} \sin \theta_1 = -\frac{h}{2} \cdot \frac{u_1(u_1 - u_0)}{\frac{1}{h} \cdot h^2 \sqrt{(u_1 - u_0)^2 + u_1^2 h^2}} \sin \theta_1 = \\ &= -\frac{u_1^2 - u_0 u_1}{2\sqrt{(u_1 - u_0)^2 + u_1^2 h^2}} \sin \theta_1 = \frac{u_0 u_1 - u_1^2}{2\sqrt{(u_1 - u_0)^2 + u_1^2 h^2}} \sin \theta_1 ; \end{aligned}$$

- $i = 1$

$$\begin{aligned}
G_1 &= \int_{\theta_0}^{\theta_2} G(\theta) d\theta = \int_{\theta_0}^{\theta_1} G(\theta) d\theta + \int_{\theta_1}^{\theta_2} G(\theta) d\theta \approx \frac{h}{2} (G(\theta_0) + 2G(\theta_1) + \\
&\quad + G(\theta_2)) = \frac{h}{2} \left(\frac{u_0 \frac{u_1 - u_0}{h} \cdot \frac{1}{h}}{\sqrt{\left(\frac{u_1 - u_0}{h}\right)^2 + u_0^2}} \sin \theta_0 + Bou_1^3 \cos \theta_1 \sin \theta_1 + \right. \\
&\quad + \frac{u_1 \frac{u_1 - u_0}{h} \cdot \frac{1}{h} + \left(2u_1^2 + \left(\frac{u_1 - u_0}{h}\right)^2\right)}{\sqrt{\left(\frac{u_1 - u_0}{h}\right)^2 + u_1^2}} \sin \theta_1 - Pu_1^4 \sin^3 \theta_1 + \lambda u_1^2 \sin \theta_1 + \\
&\quad + \frac{u_1 \frac{u_2 - u_1}{h} \cdot \left(-\frac{1}{h}\right) + \left(2u_1^2 + \left(\frac{u_2 - u_1}{h}\right)^2\right)}{\sqrt{\left(\frac{u_2 - u_1}{h}\right)^2 + u_1^2}} \sin \theta_1 + Bou_1^3 \cos \theta_1 \sin \theta_1 - \\
&\quad \left. - Pu_1^4 \sin^3 \theta_1 + \lambda u_1^2 \sin \theta_1 + \frac{u_2 \frac{u_2 - u_1}{h} \cdot \left(-\frac{1}{h}\right)}{\sqrt{\left(\frac{u_2 - u_1}{h}\right)^2 + u_2^2}} \sin \theta_2 \right) = \\
&= \frac{h}{2} \left(\frac{u_0 u_1 - u_0^2}{h \sqrt{(u_1 - u_0)^2 + h^2 u_0^2}} + \frac{2u_1^2 - 3u_0 u_1 + 2h^2 u_1^2 + u_0^2}{h \sqrt{(u_1 - u_0)^2 + u_1^2 h^2}} + \right. \\
&\quad + 2Bou_1^3 \cos \theta_1 \sin \theta_1 - 2Pu_1^4 \sin^3 \theta_1 + 2\lambda u_1^2 \sin \theta_1 + \\
&\quad \left. + \frac{2u_1^2 - 3u_1 u_2 + 2h^2 u_1^2 + u_2^2}{h \sqrt{(u_2 - u_1)^2 + u_1^2 h^2}} \sin \theta_1 + \frac{u_1 u_2 - u_2^2}{h \sqrt{(u_2 - u_1)^2 + h^2 u_2^2}} \sin \theta_2 \right) = \\
&= \frac{u_0 u_1 - u_0^2}{2 \sqrt{(u_1 - u_0)^2 + h^2 u_0^2}} + \frac{2u_1^2 - 3u_0 u_1 + 2h^2 u_1^2 + u_0^2}{2 \sqrt{(u_1 - u_0)^2 + u_1^2 h^2}} + \\
&\quad + \frac{2u_1^2 - 3u_1 u_2 + 2h^2 u_1^2 + u_2^2}{2 \sqrt{(u_2 - u_1)^2 + u_1^2 h^2}} \sin \theta_1 + \frac{u_1 u_2 - u_2^2}{2 \sqrt{(u_2 - u_1)^2 + h^2 u_2^2}} \sin \theta_2 + \\
&\quad + hBou_1^3 \cos \theta_1 \sin \theta_1 - Phu_1^4 \sin^3 \theta_1 + h\lambda u_1^2 \sin \theta_1 ;
\end{aligned}$$

- $i = \overline{1, N-1}$

$$\begin{aligned}
G_i = & \frac{u_i u_{i-1} - u_{i-1}^2}{2\sqrt{(u_i - u_{i-1})^2 + u_{i-1}^2 h^2}} \sin \theta_{i-1} + \\
& + \frac{2u_i^2 - 3u_i u_{i-1} + 2u_i^2 h^2 + u_{i-1}^2}{2\sqrt{(u_i - u_{i-1})^2 + u_i^2 h^2}} \sin \theta_i + \\
& + \frac{2u_i^2 - 3u_i u_{i+1} + 2u_i^2 h^2 + u_{i+1}^2}{2\sqrt{(u_{i+1} - u_i)^2 + u_i^2 h^2}} \sin \theta_i + \\
& + \frac{u_i u_{i+1}^2}{2\sqrt{(u_{i+1} - u_i)^2 + u_{i+1}^2 h^2}} \sin \theta_{i+1} + h B u_i^3 \sin \theta_i \cos \theta_i - \\
& - h P u_i^4 \sin^3 \theta_i + h \lambda u_i^2 \sin \theta_i ;
\end{aligned}$$

- $i = N$

$$\begin{aligned}
G_N = & \frac{u_N u_{N-1} - u_{N-1}^2}{2\sqrt{(u_N - u_{N-1})^2 + u_{N-1}^2 h^2}} \sin \theta_{N-1} + \\
& + \frac{2u_N^2 - 3u_N u_{N-1} + 2u_N^2 h^2 + u_{N-1}^2}{2\sqrt{(u_N - u_{N-1})^2 + u_N^2 h^2}} - \frac{h}{2} P u_N^4 + \frac{h}{2} \lambda u_N^2 - u_N \cos \alpha.
\end{aligned}$$

С учетом условия сохранения объема (1.11), перепишем данную формулу, аппроксимируя по формуле (2.3) и получим:

- $i = N+1$

$$G_{N+1} = 2\pi \sum_{i=1}^N \left(\frac{h}{2} \left((G(\theta_i) + G(\theta_{i-1})) \right) - 1 = \right.$$

$$\begin{aligned}
&= 2\pi \cdot \frac{h}{2} \sum_{i=1}^N \left(\frac{u_i^3}{3} \sin \theta_i + \frac{u_{i-1}^3}{3} \sin \theta_{i-1} \right) - 1 = \\
&= \frac{2\pi h}{3} \left(\sum_{i=1}^{N-1} u_i^3 \sin \theta_i + \frac{u_N^3}{2} \right) - 1.
\end{aligned}$$

В результате получим систему нелинейных алгебраических уравнений

$$\mathbf{G}(\mathbf{u}) = \begin{pmatrix} G_0(u_0, u_1) \\ G_1(u_0, u_1, u_2, \lambda, Bo, P) \\ \vdots \\ G_i(u_{i-1}, u_i, u_{i+1}, \lambda, Bo, P) \\ G_{N-1}(u_{N-2}, u_{N-1}, u_N, \lambda, Bo, P) \\ G_N(u_{N-1}, u_N, \lambda, P, \alpha) \\ G_{N+1}(u_1, \dots, u_N) \end{pmatrix}, \quad (2.4)$$

где $\mathbf{u} = (u_0, \dots, u_N, \lambda)^T$ – вектор неизвестных.

2.3 Метод Ньютона решения систем нелинейных уравнений

2.3.1 Общие сведения

Метод Ньютона, представляющий собой алгоритм, основанный на итерационном процессе, и позволяющий последовательно приближаться к корню системы, зарекомендовал себя как один из наиболее эффективных методов решения систем нелинейных уравнений.

Основной смысл заключается в использовании на каждом шаге матрицы Якоби, которая содержит частные производные всех уравнений системы по всем переменным. С помощью нее строится линейная аппроксимация системы и находится следующее приближение решения. Вычисления продолжаются, пока не будет достигнута заданная точность последовательных приближений.

Если говорить о преимуществах метода, то ключевое – это квадратичная скорость сходимости вблизи решения. За счет этого алгоритм работает значительно быстрее многих итерационных методов. Для системы

уравнений это достаточно важно, так как это дает возможность достичь более высокой точности даже при многомерных вычислениях.

В список ограничений метода входит вычисление матрицы Якоби на каждом шаге, что может быть трудоемко особенно для сложных систем. Помимо этого, сходимость очень сильно зависит от удачно выбранного начального приближения, иначе метод может расходиться.

Метод Ньютона широко применим в физике, экономике, машинном обучении для решения задач оптимизации, моделирования сложных систем и так далее. Его такая популярность обусловлена высокой точностью и скоростью работы, если имеется корректная реализация.

2.3.2 Описание метода Ньютона

Нахождение равновесной формы капли с заданными числами Бонда Bo , Вебера Re и углом смачивания α сводится к решению нелинейной системы уравнений (2.4). Для решения данной системы будет применен метод Ньютона [3].

Суть метода заключается в том, что необходимо на k -ой итерации найти вектор поправок $\Delta \mathbf{u}^k$ путем решения системы алгебраических уравнений

$$\frac{dG}{d\mathbf{u}}(\mathbf{u}^k) \Delta \mathbf{u}^k = -G(\mathbf{u}^k),$$

где $\mathbf{u}^k = (u_0^k, \dots, u_N^k, \lambda^k)^T$; $\frac{dG}{d\mathbf{u}}(\mathbf{u}^k)$ – разреженная матрица Якоби вида

$$\frac{dG}{d\mathbf{u}}(\mathbf{u}^k) = \begin{pmatrix} \frac{\partial G_0}{\partial u_0} & \frac{\partial G_0}{\partial u_1} & 0 & 0 & \dots & \dots & 0 & 0 \\ \frac{\partial G_1}{\partial u_0} & \frac{\partial G_1}{\partial u_1} & \frac{\partial G_1}{\partial u_2} & 0 & \dots & \dots & 0 & \frac{\partial G_1}{\partial \lambda} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \frac{\partial G_i}{\partial u_{i-1}} & \frac{\partial G_i}{\partial u_i} & \frac{\partial G_i}{\partial u_{i+1}} & \dots & 0 & \frac{\partial G_i}{\partial \lambda} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \frac{\partial G_{N-1}}{\partial u_{N-2}} & \frac{\partial G_{N-1}}{\partial u_{N-1}} & \frac{\partial G_{N-1}}{\partial u_N} & \frac{\partial G_{N-1}}{\partial \lambda} \\ 0 & \dots & \dots & \dots & 0 & \frac{\partial G_N}{\partial u_{N-1}} & \frac{\partial G_N}{\partial u_N} & \frac{\partial G_N}{\partial \lambda} \\ 0 & \frac{\partial G_{N+1}}{\partial u_1} & \frac{\partial G_{N+1}}{\partial u_2} & \dots & \frac{\partial G_{N+1}}{\partial u_i} & \dots & \frac{\partial G_{N+1}}{\partial u_N} & 0 \end{pmatrix}. \quad (2.5)$$

Компоненты данной матрицы расписываются следующим образом:

$$\frac{\partial G_0}{\partial u_0} = \frac{u_1^3 h^2}{2(u_1^2 h^2 + (u_0 - u_1)^2)^{\frac{3}{2}}} \sin \theta_1;$$

$$\frac{\partial G_0}{\partial u_1} = -\frac{u_1^3 h^2 - (u_0 - u_1)^3}{2(u_1^2 h^2 + (u_0 - u_1)^2)^{\frac{3}{2}}} \sin \theta_1;$$

$$\frac{\partial G_i}{\partial u_{i-1}} = \frac{-(u_{i-1}^3 h^2 - (u_i - u_{i-1})^3)}{2(u_{i-1}^2 h^2 + (u_i - u_{i-1})^2)^{\frac{3}{2}}} \sin \theta_{i-1} - \frac{u_i^3 h^2 + (u_i - u_{i-1})^3}{2(u_i^2 h^2 + (u_i - u_{i-1})^2)^{\frac{3}{2}}} \sin \theta_i,$$

$$i = \overline{1, N-1};$$

$$\frac{\partial G_i}{\partial u_i} = \frac{h^2 u_{i-1}^3 \sin \theta_{i-1}}{2(u_{i-1}^2 h^2 + (u_i - u_{i-1})^2)^{\frac{3}{2}}} + \frac{h^2 u_{i+1}^3 \sin \theta_{i+1}}{2(u_{i+1}^2 h^2 + (u_i - u_{i+1})^2)^{\frac{3}{2}}} +$$

$$+ \frac{2(h^2 + 1)^2 u_i^3 - 6(h^2 + 1)u_{i-1}u_i^2 + 3(h^2 + 2)u_i u_{i-1}^2 - 2u_{i-1}^3}{2((u_i - u_{i-1})^2 + u_i^2 h^2)^{\frac{3}{2}}} \sin \theta_i +$$

$$+ \frac{2(h^2 + 1)^2 u_i^3 - 6(h^2 + 1)u_{i+1}u_i^2 + 3(h^2 + 2)u_i u_{i+1}^2 - 2u_{i+1}^3}{2((u_i - u_{i+1})^2 + u_i^2 h^2)^{\frac{3}{2}}} \sin \theta_i +$$

$$+ h u_i \sin \theta_i (2\lambda + 3u_i B o \cos \theta_i - 4P u_i^2 \sin^2 \theta_i), \quad i = \overline{1, N-1};$$

$$\frac{\partial G_i}{\partial u_{i+1}} = -\frac{u_i^3 h^2 - (u_{i+1} - u_i)^3}{2(u_i^2 h^2 + (u_i - u_{i+1})^2)^{\frac{3}{2}}} \sin \theta_i - \frac{u_{i+1}^3 h^2 - (u_i - u_{i+1})^3}{2(u_{i+1}^2 h^2 + (u_i - u_{i+1})^2)^{\frac{3}{2}}} \sin \theta_{i+1},$$

$$i = \overline{1, N-1};$$

$$\frac{\partial G_N}{\partial u_{N-1}} = \frac{(u_N - u_{N-1})^3 - u_{N-1}^3 h^2}{2((u_N - u_{N-1})^2 + u_{N-1}^2 h^2)^{\frac{3}{2}}} \sin \theta_{N-1} + \frac{(u_{N-1} - u_N)^3 - h^2 u_N^3}{2((u_N - u_{N-1})^2 + u_N^2 h^2)^{\frac{3}{2}}};$$

$$\begin{aligned}\frac{\partial G_N}{\partial u_N} &= \frac{h^2 u_{N-1}^3}{2((u_N - u_{N-1})^2 + u_{N-1}^2 h^2)^{\frac{3}{2}}} \sin \theta_{N-1} + \\ &+ \frac{(2(h^2 + 1)^2 u_N^3 - 6(h^2 + 1)u_{N-1}u_N^2 + 3(h^2 + 2)u_N u_{N-1}^2 - 2u_{N-1}^3)}{2((u_N - u_{N-1})^2 + u_N^2 h^2)^{\frac{3}{2}}} - \\ &- 2hPu_N^3 + h\lambda u_N - \cos \alpha ;\end{aligned}$$

$$\frac{\partial G_{N+1}}{\partial u_i} = 2\pi h u_i^2 \sin \theta_i, \quad i = \overline{1, N-1};$$

$$\frac{\partial G_{N+1}}{\partial u_N} = \pi h u_N^2;$$

$$\frac{\partial G_i}{\partial \lambda} = u_i^2 h \sin \theta_i ;$$

$$\frac{\partial G_N}{\partial \lambda} = \frac{h}{2} u_N^2.$$

После нахождения вектора поправок $\Delta \mathbf{u}^k$ с помощью метода Гаусса вычислим очередное приближение:

$$\mathbf{u}^{k+1} = \Delta \mathbf{u}^k + \mathbf{u}^k.$$

В качестве начального приближения $\mathbf{u}^0$ при заданных числах Бонда и Вебера, а также угле смачивания рассматривался сферический сегмент единичного объема с заданным углом смачивания [2], т.е.

$$u(\theta) = -R \cos \alpha \cos \theta + R (1 - \cos^2 \alpha \sin^2 \theta)^{\frac{1}{2}},$$

где $R = \sqrt[3]{\frac{3}{\pi(2 + \cos^3 \alpha - 3 \cos \alpha)}}$ — безмерный радиус сферы.

Условием остановки итерационного процесса выступает следующее условие: $\|\Delta \mathbf{u}^k\|_{\infty} \leq \varepsilon, \varepsilon = 10^{-7}$. Если не наблюдалось монотонной сходимости итерационного процесса, то полагали, что метод расходится.

ГЛАВА 3

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В данном разделе будет описана программная реализация метода конечных элементов для построения равновесных форм капли. Программный код написан на языке Java, т.к. он предоставляет стандартную библиотеку, в которой уже достаточно большое количество функций реализованы, имеет высокую производительность и в целом очень прост в освоении. Все графические результаты были получены с помощью сервиса yotx.ru. Используя этот сервис удобно строить графики по заданным точкам, причем сразу несколько, что подходит для более наглядного анализа результатов.

3.1 Алгоритм нахождения равновесных форм капли

Был выбран следующий алгоритм:

1. Задаем количество разбиений $N = 500$, шаг $h = \frac{\pi}{2N}$, точность $\varepsilon = 10^{-7}$, угол α .
2. Задаем сетку методом *theta_i*.
3. Задаем числа Бонда Bo и начальное приближение.
4. Реализуем методы *systemF* для задания системы уравнений и *JacobiMatrix* для нахождения матрицы Якоби. В данном случае матрица Якоби является разреженной.
5. Решаем систему нелинейных уравнений методом Ньютона. Для него задаем начальное приближение, затем находим вектор поправок в методе *DeltaU* с помощью метода Гаусса *GaussMethod* и в итоге получаем решение, которое имеет вид $u^{k+1} = (u_0^{k+1}, \dots, u_N^{k+1}, \lambda^{k+1})^T$.
6. Если метод сошелся, то для графических результатов представляем полученное решение в цилиндрических координатах:
 - $x_i = u_i \cdot \sin \theta_i, i = \overline{0, N}$;
 - $z_i = u_i \cdot \cos \theta_i, i = \overline{0, N}$;
7. Строим графическое решение задачи.

Для удобства в процессе расчетов используем следующие методы:

- *multiply* - умножение вектора на число;
- *sum* - сложение двух чисел;
- *print* - вывод полученного решения, а также цилиндрических координат;

- *printPos* и *printNeg* - для вывода положительных и отрицательных координат соответственно для построения графиков.

Полный листинг программы на языке Java изложен в Приложении А.

3.2 Поиск критических значений числа Вебера

Для улучшения сходимости итерационного процесса при нахождении равновесных форм капли с увеличением числа Вебера и фиксированных значениях числа Бонда, а также угла смачивания был использован метод продолжения по параметру. Тем не менее наступал момент, когда с увеличением числа Вебера итерационный процесс начинал расходиться. Ниже приведен алгоритм нахождения критических значений числа Вебера, используя метод дихотомии, так как из теории известно, что с увеличением числа Вебера при фиксированных значениях числа Бонда и угла смачивания наступает момент физического кризиса равновесного состояния.

Метод продолжения по параметру реализован следующим образом:

1. Задаем число Бонда и угол смачивания α .
2. Устанавливаем начальное значение числа Вебера $P = 0$.
3. Устанавливаем шаг ΔP ;
4. Задаем границы a и b . Увеличиваем число Вебера, т.е. $P = P + \Delta P$, пока не начнет расходиться итерационный процесс. Если он расходится, то берем $a = P - \Delta P$, а $b = P$. Если при заданном числе Вебера P итерационный процесс сходится, то численное решение сохраняем в *lastGoodSolution* и затем берем его в качестве начального приближения.

Следующий за ним метод дихотомии написан по такому алгоритму:

1. Задаем точность $\varepsilon = 10^{-4}$.
2. Используем уже полученный интервал $[a, b]$ и цикл дихотомии продолжаем до тех пор, пока $b - a > \varepsilon$.
3. Считаем середину интервала $P = \frac{a+b}{2}$ и:
 - 3.1 в качестве начального приближения берем *lastGoodSolution*;
 - 3.2 если при P итерационный процесс сошелся, то $a = P$, в противном случае $b = P$.
4. Выводим посчитанное критическое значение числа Вебера P .

3.3 Поиск экспериментального порядка точности

Чтобы найти экспериментальный порядок точности воспользуемся следующими соотношениями [2]:

$$||\mathbf{u}_h - \mathbf{u}||_\infty \sim a \cdot h^p = O(h^p),$$

$$||\mathbf{u}_{2h} - \mathbf{u}||_\infty \sim a \cdot (2h)^p = O(h^p),$$

где u - точное решение задачи;

u_h - численное решение задачи с шагом h ;

u_{2h} - численное решение задачи с шагом $2h$;

p - порядок точности.

Тогда имеет место выражение для расчета порядка точности p :

$$\frac{||\mathbf{u}_{2h} - \mathbf{u}||_\infty}{||\mathbf{u}_h - \mathbf{u}||_\infty} \sim 2^p \Rightarrow p \sim \log_2 \frac{||\mathbf{u}_{2h} - \mathbf{u}||_\infty}{||\mathbf{u}_h - \mathbf{u}||_\infty}.$$

За точное решение возьмем то, которое получается на сетке при $N = 4096$. А также реализуем вычислительный эксперимент на последовательности равномерных сеток с количеством разбиений $N = 32, 64, 128, 256, 512, 1024, 2048$.

3.4 Полученные результаты

3.4.1 Графические результаты при различных определяющих параметрах

На рисунках 3.1 – 3.3 получены равновесные формы капли для различных чисел Вебера при фиксированном числе Бонда и угле смачивания. Можно заметить, что при малых значениях числа Вебера капля сохраняет выпуклую форму, а при увеличении числа Вебера капля смещается к центру плоскости.

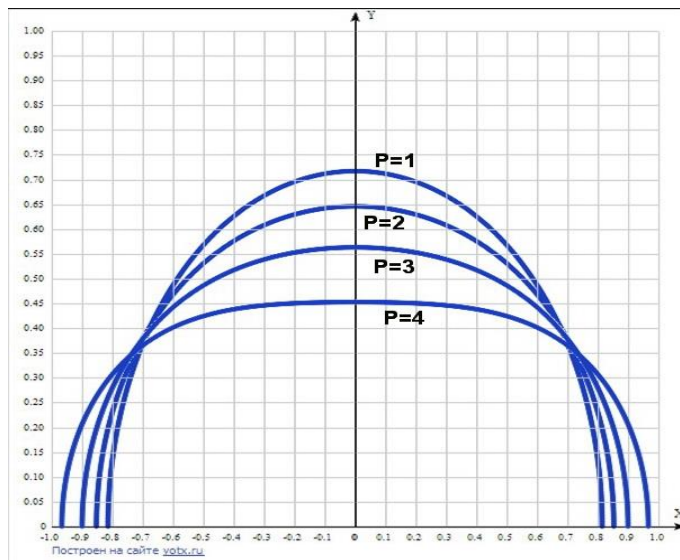


Рисунок 3.1 – Формы капли при различных P , $Bo = 0$ и $\alpha = \frac{\pi}{2}$.

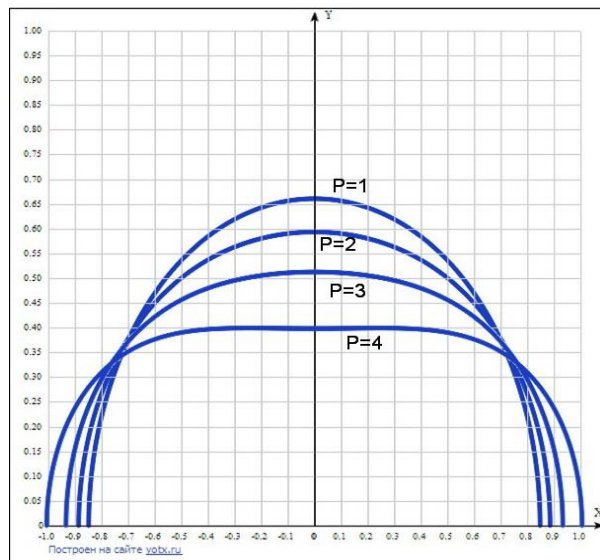


Рисунок 3.2 – Формы капли при различных P , $Bo = 1$ и $\alpha = \frac{\pi}{2}$.

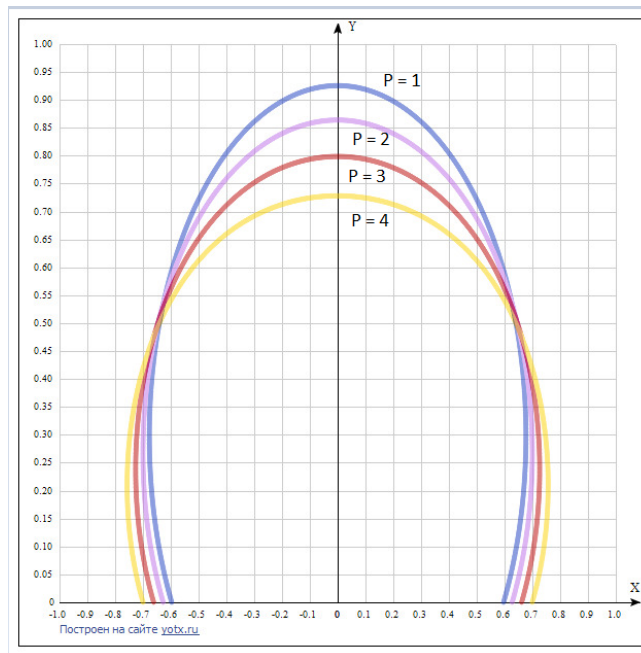


Рисунок 3.3 – Формы капли при различных P , $Bo = 0$ и $\alpha = \frac{2\pi}{3}$.

На рисунке 3.4 показаны формы капли, полученные при изменяющемся угле α и заданных числах Бонда Bo и Вебера P .

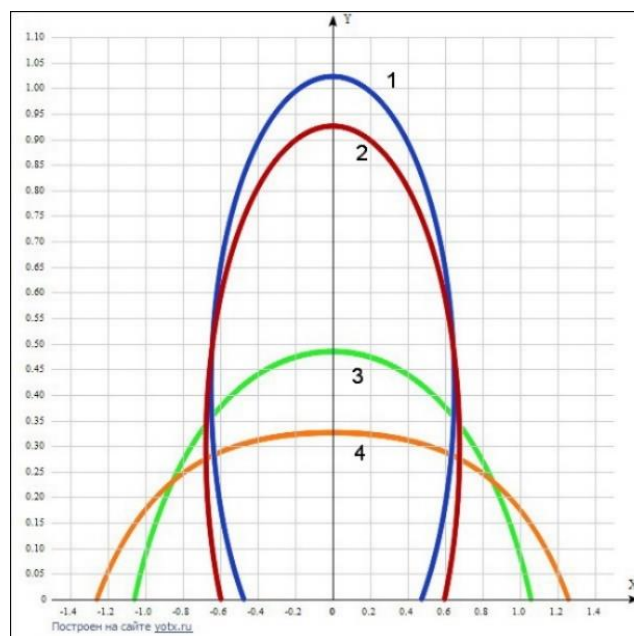


Рисунок 3.4 – Формы капли при $Bo = 1$, $P = 1$ и различных значениях угла α :

$$\alpha = \frac{2\pi}{3} (1); \frac{3\pi}{4} (2); \frac{\pi}{3} (3); \frac{\pi}{4} (4).$$

Исходя из графических результатов можно прийти к выводу, что при увеличении угла α радиус смачивания становится меньше и за счет этого капля поднимается выше.

Полученные численные результаты, по которым строились данные графики, согласуются с уже имеющимися численными результатами [2, 5].

3.4.2 Критические значения числа Вебера

В результате эксперимента, при фиксированных значениях числа Бонда Bo и угла α , с увеличением числа Вебера P наступал кризис итерационного процесса, т.к. происходит потеря физической устойчивости относительно осесимметричных возмущений. Были найдены критические значения P_{cr} . В таблице 3.1 представлены полученные результаты.

Таблица 3.1 – Критические значения числа Вебера

α	$Bo = 0$	$Bo = 1$	$Bo = 10$
$\pi / 6$	0.632	0.598	0.481
$\pi / 4$	1.384	1.323	1.086
$\pi / 3$	2.367	2.276	1.892
$\pi / 2$	4.763	4.602	3.860
$2\pi / 3$	7.206	6.963	5.839
$3\pi / 4$	8.244	7.955	6.653
$5\pi / 6$	9.061	8.727	7.272

Полученные результаты с данными критических значений согласуются с уже имеющимися численными результатами [2, 5]. Например, для $Bo = 1$ и угла смачивания $\alpha = \frac{\pi}{4}$ значение $P_{cr} = 1.323$, а для $Bo = 10$ и угла смачивания $\alpha = \frac{5\pi}{6}$ значение $P_{cr} = 7.272$.

3.4.3 Экспериментальное определение порядка точности

В таблице 3.2 приведены значения экспериментально посчитанного порядка точности для $Bo = 10$, $P = 2$ и угла $\alpha = \frac{\pi}{2}$.

Таблица 3.2 – Экспериментальный порядок точности метода при $Bo = 10$, $P = 2$ и $\alpha = \frac{\pi}{2}$.

N	32	64	128	256
$ \mathbf{u}_h - \mathbf{u} _\infty$	$2.36 \cdot 10^{-3}$	$5.84 \cdot 10^{-4}$	$1.46 \cdot 10^{-4}$	$3.63 \cdot 10^{-5}$
p	-	2.012	2.004	2.005

N	512	1024	2048
$ \mathbf{u}_h - \mathbf{u} _\infty$	$8.97 \cdot 10^{-6}$	$2.13 \cdot 10^{-6}$	$4.27 \cdot 10^{-7}$
p	2.017	2.070	2.321

В таблице 3.3 приведены значения экспериментально посчитанного порядка точности для $Bo = 1$, $P = 3$ и угла $\alpha = \frac{3\pi}{4}$.

Таблица 3.3 – Экспериментальный порядок точности метода при $Bo = 1$, $P = 3$ и $\alpha = \frac{3\pi}{4}$.

N	32	64	128	256
$ \mathbf{u}_h - \mathbf{u} _\infty$	$6.32 \cdot 10^{-4}$	$1.81 \cdot 10^{-4}$	$5.12 \cdot 10^{-5}$	$1.42 \cdot 10^{-5}$
p	-	1.733	1.776	1.810

N	512	1024	2048
$ \mathbf{u}_h - \mathbf{u} _\infty$	$3.88 \cdot 10^{-6}$	$1.01 \cdot 10^{-6}$	$2.29 \cdot 10^{-7}$
p	1.875	1.935	2.150

Все полученные результаты экспериментально посчитанного порядка точности согласуются с уже полученными результатами [2].

Из результатов видно, что метод второго порядка точности.

ЗАКЛЮЧЕНИЕ

В ходе исследования была рассмотрена вариационная постановка осесимметричной задачи о равновесных формах капли, которая находится на вращающейся горизонтальной плоскости в поле силы тяжести. Также был построен метод конечных элементов для нахождения равновесных форм свободной поверхности капли, который сводит решение задачи к системе нелинейных алгебраических уравнений. Были проведены численные расчеты: построены равновесные формы свободной поверхности капли при заданных значениях определяющих параметров, получены критические значения числа Вебера R , исследован порядок точности метода.

Предложенный в главе 3 алгоритм обеспечивает построение равновесных форм свободной поверхности капли при определяющих параметрах. Значимость данных результатов доказывается использованием их в развивающихся областях промышленности.

Итак, полученные в работе результаты согласуются с уже известными теоретическими результатами. Это подчеркивает эффективность работы метода конечных элементов и подтверждает необходимость его применения для исследования устойчивости равновесных форм капиллярных поверхностей относительно осесимметричных возмущений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Авдейчик, Е.В. Численное исследование относительного равновесия капли с односвязной свободной поверхностью на вращающейся на плоскости / Е.В.Авдейчик, П.Н.Канон // Журнал Белорусского государственного университета. Математика. Информатика. - 2022. - №3. - С.79-90.
2. Горбачева, Ю.Н. Вариационно-разностный метод численного моделирования равновесных капиллярных поверхностей / Ю.Н.Горбачева, В.К.Полевилов // Информатика. - 2023. - Т.20, №4. - С.56-68.
3. Калиткин, Н.Н. Численные методы / Н.Н. Калиткин. Главная редакция физико-математической литературы изд-ва «Наука», М., 1978
4. Метод конечных элементов [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Метод_конечных_элементов /–Дата доступа: 20.05.2025
5. Мышкис, А.Д. Методы решения задач гидромеханики для условий невесомости / А.Д.Мышкис; Киев : Наукова думка, 1992. - 592 с.
6. Полевилов, В.К. Численное исследование равновесных форм капли, вращающейся в гравитационном поле / В.К.Полевилов, В.М.Денисенко // Вестник Белорусского государственного университета имени В.И.Ленина. - 1985. - №2. - С.37-41.
7. Солдаткин, А.В. Введение в метод конечных элементов: учебное пособие / А.В. Солдаткин, Е.С. Баранова; Балт. гос. техн. ун-т. – СПб., 2020. – 123 с
8. Финн, Р. Равновесные капиллярные поверхности. Математическая теория : пер. с англ. / Р. Финн. – М. : Мир, 1989. – 312 с.
9. Dingle, N. M. A robust algorithm for the simultaneous parameter estimation of interfacial tension and contact angle from sessile drop profiles / N. M. Dingle, M. T. Harris // J. of Colloid and Interface Science. – 2005. – Vol. 286, no. 2. – P. 670–680. <https://doi.org/10.1016/j.jcis.2005.01.087>

ПРИЛОЖЕНИЕ А

Листинг А.1 – алгоритм построения равновесных форм капли и нахождения критических значений числа Вебера

```
import java.io.*;
import java.text.DecimalFormat;
import java.util.Arrays;
import java.util.Locale;
import java.util.function.Function;

public class Main {
    static int  $N = 500$ ; // число разбиений
    static double  $PI = \text{Math.PI}$ ;
    static double  $h = PI / (2 * N)$ ;
    static double  $\epsilon = \text{Math.pow}(10, -7)$ ;
    static double  $\alpha = PI / 2$ ;
    static double  $P = 0.0$ ;
    static double  $Bo = 0.0$ ;

    public static double norm(double[] u) {
        double max = Double.NEGATIVE_INFINITY;
        for (double value : u) {
            double absValue = Math.abs(value);
            if (absValue > max) {
                max = absValue;
            }
        }
        return max;
    }

    public static double theta_i(int i) {
        return ( $\text{Math.PI} * i$ ) / ( $2 * N$ );
    }

    static double[] systemF(double[] u, double h, double P, double Bo) { //alpha
        было
        double[] F = new double[N + 2];
        F[0] = (u[0] * u[1] - Math.pow(u[1], 2)) * Math.sin(theta_i(1)) / (2.0 *
            Math.sqrt(Math.pow(u[1] - u[0], 2) + Math.pow(u[1], 2) * Math.pow(h, 2)));
    }
}
```

```

    for (int i = 1; i < N; i++) {
        F[i] = h * Bo * Math.pow(u[i], 3) * Math.cos(theta_i(i)) *
Math.sin(theta_i(i)) +
            h * u[N + 1] * Math.pow(u[i], 2) * Math.sin(theta_i(i)) -
            h * P * Math.pow(u[i], 4) * Math.pow(Math.sin(theta_i(i)), 3) +
            (u[i] * u[i - 1] - Math.pow(u[i - 1], 2)) * Math.sin(theta_i(i - 1)) / (2.0
* Math.sqrt(Math.pow(u[i] - u[i - 1], 2) + Math.pow(u[i - 1], 2) * Math.pow(h, 2)))
+
            (2 * Math.pow(u[i], 2) - 3 * u[i] * u[i - 1] + 2 * Math.pow(u[i], 2) *
Math.pow(h, 2) + Math.pow(u[i - 1], 2)) * Math.sin(theta_i(i)) / (2.0 *
Math.sqrt(Math.pow(u[i] - u[i - 1], 2) + Math.pow(u[i], 2) * Math.pow(h, 2))) +
            (2 * Math.pow(u[i], 2) - 3 * u[i] * u[i + 1] + 2 * Math.pow(u[i], 2) *
Math.pow(h, 2) + Math.pow(u[i + 1], 2)) * Math.sin(theta_i(i)) / (2.0 *
Math.sqrt(Math.pow(u[i + 1] - u[i], 2) + Math.pow(u[i], 2) * Math.pow(h, 2))) +
            (u[i] * u[i + 1] - Math.pow(u[i + 1], 2)) * Math.sin(theta_i(i + 1)) /
(2.0 * Math.sqrt(Math.pow(u[i + 1] - u[i], 2) + Math.pow(u[i + 1], 2) *
Math.pow(h, 2)));

    }

    F[N] = (u[N] * u[N - 1] - Math.pow(u[N - 1], 2)) * Math.sin(theta_i(N - 1)) /
(2.0 * Math.sqrt(Math.pow(u[N] - u[N - 1], 2) + Math.pow(u[N - 1], 2) *
Math.pow(h, 2))) +
        (2 * Math.pow(u[N], 2) - 3 * u[N] * u[N - 1] + 2 * Math.pow(u[N], 2) *
Math.pow(h, 2) + Math.pow(u[N - 1], 2)) / (2.0 * Math.sqrt(Math.pow(u[N] - u[N -
1], 2) + Math.pow(u[N], 2) * Math.pow(h, 2))) -
        0.5 * h * P * Math.pow(u[N], 4) + 0.5 * h * u[N + 1] * Math.pow(u[N],
2) - u[N] * Math.cos(alpha);

    double sum = 0;
    for (int i = 1; i < N; i++) {
        sum += Math.pow(u[i], 3) * Math.sin(theta_i(i));
    }

    F[N + 1] = -1 + (2.0 * PI * h / 3) * (sum + 0.5 * Math.pow(u[N], 3));
    return F;
}

```

```

static double[][] JacobiMatrix(double[] u, double h, double P, double Bo) {
    double[][] M = new double[N + 2][N + 2];
    for (int i = 0; i < N + 2; i++) {
        for (int j = 0; j < N + 2; j++) {

```

```

    M[i][j] = 0;
  }
}

```

```

M[0][0] = (Math.pow(h, 2) * Math.pow(u[1], 3) * Math.sin(theta_i(1))) / (2.0
* (Math.pow(Math.pow(u[0] - u[1], 2) + Math.pow(u[1], 2) * Math.pow(h, 2), 3.0 /
2.0)));

```

```

M[0][1] = (Math.pow(u[0] - u[1], 3) - Math.pow(u[1], 3) * Math.pow(h, 2)) *
Math.sin(theta_i(1)) / (2.0 * Math.pow(Math.pow(u[0] - u[1], 2) + Math.pow(u[1],
2) * Math.pow(h, 2), 3.0 / 2.0));

```

```

for (int i = 1; i < N; i++) {
    M[i][i - 1] = (Math.pow(u[i] - u[i - 1], 3) - Math.pow(u[i - 1], 3) *
Math.pow(h, 2)) * Math.sin(theta_i(i - 1)) / (2.0 * Math.pow(Math.pow(u[i] - u[i -
1], 2) + Math.pow(u[i - 1], 2) * Math.pow(h, 2), 3.0 / 2.0)) +
    (Math.sin(theta_i(i)) * (Math.pow(u[i - 1] - u[i], 3) - Math.pow(u[i], 3)
* Math.pow(h, 2))) / (2.0 * Math.pow(Math.pow(u[i] - u[i - 1], 2) + Math.pow(u[i],
2) * Math.pow(h, 2), 3.0 / 2.0));
    M[i][i] = (Math.pow(h, 2) * Math.pow(u[i - 1], 3)) * Math.sin(theta_i(i -
1)) / (2 * Math.pow(Math.pow(h, 2) * Math.pow(u[i - 1], 2) + Math.pow(u[i] - u[i -
1], 2), 3.0 / 2.0)) +
    (Math.pow(h, 2) * Math.pow(u[i + 1], 3)) * Math.sin(theta_i(i + 1)) /
(2.0 * Math.pow(Math.pow(h, 2) * Math.pow(u[i + 1], 2) + Math.pow(u[i] - u[i +
1], 2), 3.0 / 2.0)) +
    (2 * Math.pow(h * h + 1, 2) * Math.pow(u[i], 3) - 6.0 * (Math.pow(h,
2) + 1) * u[i - 1] * Math.pow(u[i], 2) + 3.0 * (Math.pow(h, 2) + 2) * u[i] *
Math.pow(u[i - 1], 2) - 2.0 * Math.pow(u[i - 1], 3)) * Math.sin(theta_i(i)) / (2 *
Math.pow(Math.pow(h, 2) * Math.pow(u[i], 2) + Math.pow(u[i] - u[i - 1], 2), 3.0 /
2.0)) +
    (2 * Math.pow(Math.pow(h, 2) + 1, 2) * Math.pow(u[i], 3) - 6.0 *
(Math.pow(h, 2) + 1) * u[i + 1] * Math.pow(u[i], 2) + 3.0 * (Math.pow(h, 2) + 2) *
u[i] * Math.pow(u[i + 1], 2) - 2.0 * Math.pow(u[i + 1], 3)) * Math.sin(theta_i(i)) /
(2.0 * Math.pow(Math.pow(h, 2) * Math.pow(u[i], 2) + Math.pow(u[i] - u[i + 1],
2), 3.0 / 2.0)) +
    3.0 * h * Bo * Math.pow(u[i], 2) * Math.sin(theta_i(i)) *
Math.cos(theta_i(i)) -
    4.0 * h * P * Math.pow(u[i], 3) * Math.pow(Math.sin(theta_i(i)), 3) +
    2.0 * h * u[N + 1] * u[i] * Math.sin(theta_i(i));
    M[i][i + 1] = (Math.pow(u[i + 1] - u[i], 3) - Math.pow(u[i], 3) *
Math.pow(h, 2)) * Math.sin(theta_i(i)) / (2.0 * Math.pow(Math.pow(u[i] - u[i + 1],
2) + Math.pow(u[i], 2) * Math.pow(h, 2), 3.0 / 2.0)) +
    (Math.pow(u[i] - u[i + 1], 3) - Math.pow(u[i + 1], 3) * Math.pow(h, 2))
* Math.sin(theta_i(i + 1)) / (2.0 * Math.pow(Math.pow(u[i] - u[i + 1], 2) +

```

```

Math.pow(u[i + 1], 2) * Math.pow(h, 2), 3.0 / 2.0));
    M[i][N + 1] = h * Math.pow(u[i], 2) * Math.sin(theta_i(i));
}

for (int i = 1; i < N; i++) {
    M[N + 1][i] = 2 * PI * h * Math.pow(u[i], 2) * Math.sin(theta_i(i));
}

M[N][N - 1] = (Math.pow(u[N] - u[N - 1], 3) - Math.pow(h, 2) *
Math.pow(u[N - 1], 3)) * Math.sin(theta_i(N - 1)) / (2.0 * Math.pow(Math.pow(h,
2) * Math.pow(u[N - 1], 2) + Math.pow(u[N] - u[N - 1], 2), 3.0 / 2.0)) +
    (Math.pow(u[N - 1] - u[N], 3) - Math.pow(h, 2) * Math.pow(u[N], 3)) /
(2.0 * Math.pow(Math.pow(h, 2) * Math.pow(u[N], 2) + Math.pow(u[N] - u[N - 1],
2), 3.0 / 2.0));
M[N][N] = (Math.pow(h, 2) * Math.pow(u[N - 1], 3)) * Math.sin(theta_i(N -
1)) / (2.0 * Math.pow(Math.pow(u[N] - u[N - 1], 2) + Math.pow(u[N - 1], 2) *
Math.pow(h, 2), 3.0 / 2.0)) +
    (2.0 * Math.pow(h * h + 1, 2) * Math.pow(u[N], 3) - 6.0 * (Math.pow(h,
2) + 1) * u[N - 1] * Math.pow(u[N], 2) + 3.0 * (Math.pow(h, 2) + 2) * u[N] *
Math.pow(u[N - 1], 2) - 2.0 * Math.pow(u[N - 1], 3)) / (2.0 *
Math.pow(Math.pow(u[N] - u[N - 1], 2) + Math.pow(u[N], 2) * Math.pow(h, 2),
3.0 / 2.0)) -
    2.0 * h * P * Math.pow(u[N], 3) +
    h * u[N + 1] * u[N] -
    Math.cos(alpha);
M[N][N + 1] = h * Math.pow(u[N], 2) / 2;
M[N + 1][N] = PI * h * Math.pow(u[N], 2);

return M;
}

public static double[] GaussMethod(double[][] A, double[] b) {
    int n = A.length;
    double[][] augmentedMatrix = new double[n][n + 1];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            augmentedMatrix[i][j] = A[i][j];
        }
        augmentedMatrix[i][n] = b[i];
    }

    // Прямой ход метода Гаусса
    for (int k = 0; k < n - 1; k++) {

```

```

        int maxIndex = k;
        for (int i = k + 1; i < n; i++) {
            if (Math.abs(augmentedMatrix[i][k]) >
Math.abs(augmentedMatrix[maxIndex][k])) {
                maxIndex = i;
            }
        }

        if (maxIndex != k) {
            double[] temp = augmentedMatrix[k];
            augmentedMatrix[k] = augmentedMatrix[maxIndex];
            augmentedMatrix[maxIndex] = temp;
        }

        for (int i = k + 1; i < n; i++) {
            double factor = augmentedMatrix[i][k] / augmentedMatrix[k][k];
            for (int j = k + 1; j < n + 1; j++) {
                augmentedMatrix[i][j] -= factor * augmentedMatrix[k][j];
            }
        }
    }
}

```

```

// Обратный ход метода Гаусса
double[] x = new double[n];
for (int k = n - 1; k >= 0; k--) {
    double sum = 0.0;
    for (int j = k + 1; j < n; j++) {
        sum += augmentedMatrix[k][j] * x[j];
    }
    x[k] = (augmentedMatrix[k][n] - sum) / augmentedMatrix[k][k];
}

return x;
}

```

```

public static double[] DeltaU(double[] u, double h, double Bo, double P) {
    return GaussMethod(JacobiMatrix(u, h, P, Bo), multiply(-1, systemF(u, h, P,
Bo)));
}

```

```

public static double[] multiply(double betta, double[] u) {
    double[] result = new double[u.length];

```

```

        for (int i = 0; i < result.length; i++) {
            result[i] = u[i] * betta;
        }
        return result;
    }

    public static double[] sum(double[] x, double[] y) { // cyMMA
        double[] result = new double[x.length];
        for (int i = 0; i < result.length; i++) {
            result[i] = x[i] + y[i];
        }
        return result;
    }

    private static void print(double[] vector, String name, int n) {
        System.out.print(name + " = (");
        for (int i = 0; i < vector.length; i++) {
            System.out.printf(Locale.US, "%. " + n + "f", vector[i]);
            if (i < vector.length - 1) {
                System.out.print(", ");
            }
        }
        System.out.println(")");
    }

    private static void printPos(double[] vector1, double[] vector2, String name, int
n) {
        StringBuilder result = new StringBuilder(name);
        for (int i = vector1.length - 1; i >= 0; i--) {
            result.append("(")
                .append(String.format(Locale.US, "%. " + n + "f", vector1[i]))
                .append(" ; ")
                .append(String.format(Locale.US, "%. " + n + "f", vector2[i]))
                .append(")");
            if (i != vector1.length - 1) {
                result.append(" ");
            }
        }
        System.out.println(result.toString());
    }

    private static void printNeg(double[] vector1, double[] vector2, int n) {
        StringBuilder result = new StringBuilder();

```

```

for (int i = 1; i < vector1.length; i++) {
    result.append("(");

    // Форматируем первое число с учетом знака
    if (vector1[i] < 0) {
        result.append(String.format(Locale.US, "%." + n + "f", vector1[i]));
    } else {
        result.append("-").append(String.format(Locale.US, "%." + n + "f",
vector1[i]));
    }

    result.append(" ; ")
        .append(String.format(Locale.US, "%." + n + "f", vector2[i]))
        .append(")");

    if (i != vector1.length - 1) {
        result.append(" ");
    }
}

System.out.println(result.toString());
}

```

```

private static double[] difference(double[] a, double[] b) {
    double[] result = new double[a.length];
    for (int i = 0; i < a.length; i++) {
        result[i] = a[i] - b[i];
    }
    return result;
}

```

```

public static void main(String[] args) {
    double R = Math.pow((3.0 / (PI * (2.0 - 3.0 * Math.cos(alpha) +
Math.pow(Math.cos(alpha), 3)))), 1.0 / 3.0);
    double[] u0 = new double[N + 2];

    for (int i = 0; i < N + 2; i++) {
        u0[i] = R * Math.sqrt(1.0 - Math.pow(Math.cos(alpha), 2) *
Math.pow(Math.sin(theta_i(i)), 2)) - R * Math.cos(alpha) * Math.cos(theta_i(i));
        u0[N + 1] = 0;
    }
}

```

```

double tol = 0.0001; // точность определения P_кр
double criticalP = P;
double[] lastGoodSolution = u0.clone(); // последнее сходящееся решение
double deltaP = 0.1;

// продолжение по параметру

double a = P;
double b = P + deltaP;
boolean intervalFound = false;

while (!intervalFound) {
    System.out.printf("Проверяем P = %.2f: ", b);
    int iter=0;

    boolean converged = false;
    double[] currentSolution = lastGoodSolution.clone();
    double[] prevDeltaU = null;

    // метод Ньютона
    while (true) {
        double[] deltaU = DeltaU(currentSolution, h, Bo, b);
        double[] nextSolution = sum(currentSolution, deltaU);

        // критерий сходимости
        if (norm(deltaU) < epsilon) {
            converged = true;
            break;
        }

        // критерий расходимости
        if (prevDeltaU != null && norm(deltaU) >= norm(prevDeltaU)) {
            break;
        }

        prevDeltaU = deltaU.clone();
        currentSolution = nextSolution;
        iter++;
    }

    if (converged) {
        System.out.printf("Сходится (%d итер.), norm_u = %.6f\n ", iter,
norm(currentSolution));
        a = b;
    }
}

```

```

lastGoodSolution = currentSolution.clone();
b += deltaP;
print(currentSolution, "u{k}", 10);
System.out.print("\n");
double[] x = new double[N + 1];
double[] z = new double[N + 1];
for (int i = 0; i < N + 1; i++) {
    x[i] = currentSolution[i] * Math.sin(theta_i(i));
    z[i] = currentSolution[i] * Math.cos(theta_i(i));
}
print(x, "X", 4); // цилиндрические координаты
print(z, "Z", 4);
System.out.print("\n");

printPos(x, z, "Координаты для графика:", 4);
printNeg(x, z, 4);
System.out.print("\n");
System.out.print("z_0 == " + z[0] + "\n");
System.out.print("x_N == " + x[N]);
System.out.print("\n");
System.out.println("-----");
} else {
    System.out.println("Расходится - интервал найден");
    intervalFound = true;
}

iter++;
}

```

```

// дихотомия с продолжением по параметру
System.out.println("\nЭтап дихотомии:");
System.out.println("Начальный интервал: [" a ", " b "]");

while (b - a > tol) {
    double mid = (a + b) / 2;
    System.out.printf("Проверяем P = %.6f: ", mid);

    boolean converged = false;
    double[] currentSolution = lastGoodSolution.clone();
    double[] prevDeltaU = null;

    // метод Ньютона с продолжением по параметру
    while (true) {

```

```

double[] deltaU = DeltaU(currentSolution, h, Bo, mid);
double[] nextSolution = sum(currentSolution, deltaU);

// критерий сходимости
if (norm(deltaU) < epsilon) {
    converged = true;
    break;
}

// критерий расходимости
if (prevDeltaU != null && norm(deltaU) >= norm(prevDeltaU)) {
    break;
}

prevDeltaU = deltaU.clone();
currentSolution = nextSolution;
}

if (converged) {
    System.out.println("Сходится");
    a = mid;
    lastGoodSolution = currentSolution.clone();
} else {
    System.out.println("Расходится");
    b = mid;
}

criticalP = (a + b) / 2;
System.out.printf("Новый интервал: [%.6f, %.6f]\n", a, b);
}

System.out.println("\nКритическое значение P_cr = " + criticalP);

}
}

```

```

public class Main {

    public static double norm(double[] u) {
        double max = Double.NEGATIVE_INFINITY;
        for (double value : u) {
            double absValue = Math.abs(value);
            if (absValue > max) {
                max = absValue;
            }
        }
        return max;
    }

    public static double findNorm(double[] vector1, double[] vector2) {
        int scalingFactor = getScalingFactor(vector1.length);
        double[] res = new double[Math.min(vector1.length, vector2.length /
scalingFactor)];

        for (int i = 0; i < res.length; i++) {
            res[i] = Math.abs(vector1[i] - vector2[i * scalingFactor]);
        }

        return norm(res);
    }

    private static int getScalingFactor(int length) {
        if (length > 2000) return 2;
        if (length > 1000) return 4;
        if (length > 500) return 8;
        if (length > 200) return 16;
        if (length > 100) return 32;
        if (length > 50) return 64;
        return 128;
    }

    public static void main(String[] args) {
        double accurate[] = {...};
        double sol_2048[]={...};
        double sol_1024[]={...};
        double sol_512[]={...}
    }
}

```

```

double sol_256[]={...};
double sol_128[]={...};
double sol_64[]={...};

double[] norms = new double[7];
norms[0] = findNorm(sol_2048, accurate);
norms[1] = findNorm(sol_1024, accurate);
norms[2] = findNorm(sol_512, accurate);
norms[3] = findNorm(sol_256, accurate);
norms[4] = findNorm(sol_128, accurate);
norms[5] = findNorm(sol_64, accurate);
norms[6] = findNorm(sol_32, accurate);

for (int i = 0; i < norms.length; i++) {
    System.out.println("Норма при N=" + (2048 / (1 << i)) + ": " + norms[i]);
}

for (int i = 1; i < norms.length; i++) {
    double p = Math.log10(norms[i] / norms[i - 1]) / Math.log10(2);
    System.out.println("Коэффициент p при N=" + (2048 / (1 << (i - 1))) +
": " + p);
}
}
}

```