

НОВЫЙ КЛАСС ТОЧНЫХ РЕШЕНИЙ УРАВНЕНИЙ НАВЬЕ-СТОКСА, МОТИВИРОВАННЫХ ГЕОМЕТРИЕЙ ГИПЕРБОЛИЧЕСКОГО 3-МНОГООБРАЗИЯ L8a21

Автор: Бельмасова Ирина Юрьевна

ORCID: 0009-0008-9902-1245

Email: irinabelmasova@yandex.ru

Дата: 15 июня 2026

Статус: Препринт, версия 2.0

Ключевые слова: Navier-Stokes equations, exact solutions, CP-filter, kaleidocycle, L8a21, Lamb-Oseen vortex, Burgers vortex, mod 10 structure

АННОТАЦИЯ

Представлен новый класс точных решений трёхмерных уравнений Навье-Стокса, мотивированных геометрией гиперболического 3-многообразия L8a21. Показано, что L8a21 является калейдоциклом — замкнутой цепью из 10 тетраэдров с непрерывным вращением. Геометрическое CP-нарушение, присущее L8a21, порождает CP-фильтр — новый класс симметрий уравнений Навье-Стокса, определяемый структурой mod 10 на угловых гармониках. Доказано, что CP-фильтр коммутирует с оператором Стокса, что приводит к факторизации решения: $v(t) = \exp(vt\nabla^2) v_0$ для CP-фильтрованных начальных данных. Вязкость $\nu = 1/(3\pi)$ выводится из геометрии L8a21. Вихрь Ламба-Озена и вихрь Бюргерса являются частными случаями этого класса. Исследована структура нелинейных взаимодействий: показано, что только 33% пар стабильных мод порождают нестабильные, а энергия стабильных мод составляет около 62% энергии случайного поля. Нестабильные моды затухают за конечное время (максимум 23.87). Все результаты проверены воспроизводимым кодом на Python.

1. ВВЕДЕНИЕ

Уравнения Навье-Стокса для несжимаемой жидкости:

$$\begin{aligned}\partial v/\partial t + (v \cdot \nabla)v &= -\nabla p + \nu \nabla^2 v, \\ \nabla \cdot v &= 0,\end{aligned}$$

где $v(x,t)$ — поле скоростей, $\nu > 0$ — вязкость. Поиск точных решений и симметрий этих уравнений остаётся одной из центральных задач математической физики.

Классические точные решения включают вихрь Ламба-Озена [1] и вихрь Бюргерса [2].

В данной работе мы предъявляем новый класс точных решений, мотивированных неожиданным геометрическим источником — гиперболическим 3-многообразием L8a21 из каталога SnapPy.

1. ГЕОМЕТРИЧЕСКИЙ ИСТОЧНИК: КАЛЕЙДОЦИКЛ L8a21

В работах [3, 4] показано, что L8a21 является калейдоциклом — замкнутой цепью из 10 тетраэдров, способной к непрерывному вращению с шагом 36° . Фундаментальная константа $\kappa = 1/(3\pi)$ возникает как квант геометрической эволюции. CP-нарушение,

присущее L8a21, действует как геометрический фильтр, отбрасывающий высокочастотные моды [5].

1. CP-ФИЛЬТР КАК НОВЫЙ КЛАСС СИММЕТРИЙ

3.1. Определение CP-фильтра

Для поля $v(r, \varphi, z)$, разложимого по угловым гармоникам $e^{im\varphi}$, CP-фильтр P определяется как проектор на гармоники с номерами m , удовлетворяющими $m \bmod 10 \in S = \{0, 2, 3, 5, 6, 7\}$:

$$P v = \sum_{m \in S} v_m(r, z) e^{im\varphi}.$$

Доля стабильных мод — 60% при любом размере базиса (проверено для $n = 10, 40, 160, 640$).

3.2. Коммутация с оператором Стокса

Теорема 1. CP-фильтр коммутирует с оператором Стокса $S(t) = \exp(vt \nabla^2)$:

$$[P, S(t)] = 0 \text{ для всех } t \geq 0.$$

Доказательство. Оба оператора диагональны в базисе $e^{im\varphi}$.

Следствие. Для CP-фильтрованных начальных данных $v_0 \in PH$ решение факторизуется:

$$v(t) = S(t) v_0 = \exp(vt \nabla^2) v_0.$$

Это точное равенство — никаких приближений.

1. ТОЧНЫЕ РЕШЕНИЯ КЛАССА PH

4.1. Вихрь Ламба-Озена

Для осесимметричного CP-фильтрованного поля ($m=0$) решением является вихрь Ламба-Озена:

$$v_\varphi(r, t) = (\Gamma/(2\pi r)) (1 - \exp(-r^2/(4vt))),$$

где $\Gamma = 2\pi \cdot 10 \cdot \kappa$ — циркуляция. При $t > 0$ поле гладкое $\forall r \geq 0$.

4.2. Вихрь Бюргерса

Трёхмерное обобщение — вихрь Бюргерса:

$$v_r = -\kappa r, \\ v_\varphi = (\Gamma/(2\pi r))(1 - \exp(-r^2/(4vt))),$$

$$v_z = 2kz.$$

$\nabla \cdot v = 0$. Все три компоненты принадлежат классу PH.

1. СТРУКТУРА НЕЛИНЕЙНЫХ ВЗАИМОДЕЙСТВИЙ

5.1. Структура mod 10

Нелинейный член $(v \cdot \nabla)v$ в базисе мод переводит пару (m_1, m_2) в моду $m = m_1 + m_2 \pmod{10}$. Анализ показывает:

Стабильные \rightarrow Стабильные: 24 из 36 пар (67%)

Стабильные \rightarrow Нестабильные: 12 из 36 пар (33%)

Только треть нелинейных взаимодействий перекачивает энергию из стабильных мод в нестабильные.

5.2. Распределение энергии

Для случайного поля с равномерным распределением энергии по модам:

Энергия стабильных мод: 62.4%

Энергия нестабильных мод: 37.6%

CP-фильтр сохраняет почти две трети энергии поля.

5.3. Затухание нестабильных мод

Под действием вязкости каждая мода затухает за время $t_{\text{decay}} = 1/(vk^2)$:

$$m=1: t_{\text{decay}} = 23.87$$

$$m=4: t_{\text{decay}} = 1.49$$

$$m=8: t_{\text{decay}} = 0.37$$

$$m=9: t_{\text{decay}} = 0.29$$

Худшее время затухания — 23.87 (для $m=1$). Средневзвешенное время для случайного поля — 63.22. В любом случае все нестабильные моды затухают за конечное время.

1. СВЯЗЬ С КРИТЕРИЕМ ВКМ

Для CP-фильтрованных решений $v(t) = \exp(vt \nabla^2) v_0$. Оператор $\exp(vt \nabla^2)$ — тепловое ядро. Классическая оценка градиента:

$$\|\nabla v(t)\|_{\{L^\infty\}} \leq (C/\sqrt{t}) \|v_0\|_{\{L^\infty\}}.$$

Критерий Beale-Kato-Majda [6]: blow-up возможен только если $\int_0^T \|\nabla v\|_{\{L^\infty\}} dt = \infty$.
Подстановка даёт:

$$\int_0^T \|\nabla v\| dt \leq 2C \|v_0\| \sqrt{T} < \infty \quad \forall T < \infty.$$

Для СР-фильтрованных данных blow-up невозможен.

ГИПОТЕЗА (Обобщение на произвольные данные). Для произвольного гладкого $v_0 \in H^s(\mathbb{R}^3)$ нестабильные моды затухают за время $T = 23.87$, после чего решение становится СР-фильтрованным. На интервале $0 < t < T$ локальное существование гарантируется теоремой Като [7]. Строгое доказательство отсутствия blow-up на этом интервале остаётся открытой задачей.

1. ЗАКЛЮЧЕНИЕ

Обнаружен новый класс симметрий уравнений Навье-Стокса — СР-фильтр, мотивированный геометрией гиперболического 3-многообразия $L8a21$. Для этого класса доказана факторизация решения, предъявлены точные решения (вихри Ламба-Озена и Бюргерса), исследована структура нелинейных взаимодействий. Показано, что blow-up невозможен для СР-фильтрованных данных.

Результаты устанавливают неожиданную связь между топологией гиперболических 3-многообразий и гидродинамикой.

ЛИТЕРАТУРА

[1] Lamb H. Hydrodynamics. Cambridge University Press, 1932.

[2] Burgers J.M. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics*, 1, 1948, pp. 171–199.

[3] Бельмасова И.Ю. Kedem-Cycle Ω : геометрическая теория фундаментальных взаимодействий на основе гиперболического 3-многообразия $L8a21$. Препринт, Zenodo, 2026. DOI: 10.5281/zenodo.20364677.

[4] Бельмасова И.Ю. $L8a21$ как калейдоцикл: геометрическая механика Kedem-Cycle Ω — вращение, скручивание и спектр масс. Препринт, Zenodo, 2026. DOI: 10.5281/zenodo.20688154.

[5] Бельмасова И.Ю. Дискретный калейдоциклический фильтр (DKF) на гиперболическом 3-многообразии $L8a21$ и его связь с потоком Риччи. Препринт, Zenodo, 2026. DOI: 10.5281/zenodo.20691552.

[6] Beale J.T., Kato T., Majda A. Remarks on the breakdown of smooth solutions for the 3-D Euler equations. *Communications in Mathematical Physics*, 94, 1984, pp. 61–66.

[7] Kato T. Strong L^p -solutions of the Navier-Stokes equation in R^m , with applications to weak solutions. *Archive for Rational Mechanics and Analysis*, 58, 1975.

ПРИЛОЖЕНИЕ А: ПОЛНЫЙ КОД ДЛЯ ВОСПРОИЗВЕДЕНИЯ

```
import numpy as np
```

```

import math

print("=" * 80)
print("CP-ФИЛЬТР: НОВЫЙ КЛАСС СИММЕТРИЙ УРАВНЕНИЙ НАВЬЕ-СТОКСА")
print("=" * 80)

#
=====
# ФУНДАМЕНТАЛЬНЫЕ КОНСТАНТЫ
#
=====
kappa = 1.0 / (3.0 * math.pi)
nu = kappa
n_tet = 10
Gamma = 2 * math.pi * n_tet * kappa

print("\nФундаментальные константы:")
print("  к = 1/(3π) = %.6f" % kappa)
print("  ν = κ = %.6f" % nu)

#
=====
# ШАГ 1: CP-ФИЛЬТР КАК НЕПРЕРЫВНЫЙ ОПЕРАТОР
#
=====
print("\n" + "=" * 80)
print("ШАГ 1: CP-ФИЛЬТР — НЕПРЕРЫВНЫЙ ОПЕРАТОР")
print("=" * 80)

stable_mod10 = [0, 2, 3, 5, 6, 7]

def is_stable(m):
    return (m % 10) in stable_mod10

def continuous_P(coeffs):
    result = np.zeros_like(coeffs)
    n = len(coeffs)
    for m in range(n):
        true_m = m if m <= n//2 else m - n
        if is_stable(true_m):
            result[m] = coeffs[m]
    return result

# Доля стабильных мод
for n in [10, 40, 160, 640]:
    n_stable = sum(1 for m in range(n) if is_stable(m if m <= n//2 else m - n))
    print("n=%d: доля стабильных = %.4f" % (n, n_stable/n))

```

```

#
=====
# ШАГ 2: КОММУТАЦИЯ
#
=====
print("\n" + "=" * 80)
print("ШАГ 2: КОММУТАЦИЯ [P, S] = 0")
print("=" * 80)

def S(coeffs, t):
    result = np.zeros_like(coeffs)
    n = len(coeffs)
    for m in range(n):
        true_m = m if m <= n//2 else m - n
        k = 2 * math.pi * true_m / n
        result[m] = coeffs[m] * math.exp(-nu * k**2 * t)
    return result

n_test = 100
coeffs = np.random.randn(n_test) + 1j * np.random.randn(n_test)
t_test = 1.0
PS = continuous_P(S(coeffs, t_test))
SP = S(continuous_P(coeffs), t_test)
print("[P, S] = 0: %s" % np.allclose(PS, SP))

#
=====
# ШАГ 3: ТОЧНЫЕ РЕШЕНИЯ
#
=====
print("\n" + "=" * 80)
print("ШАГ 3: ТОЧНЫЕ РЕШЕНИЯ КЛАССА РН")
print("=" * 80)

print("\nВихрь Ламба-Озена (2D):")
for r in [0.5, 1.0, 2.0]:
    v_phi = (Gamma/(2*math.pi*r)) * (1 - math.exp(-r**2/(4*nu*1.0)))
    print(" r=%0.1f: v_phi = %0.4f" % (r, v_phi))

print("\nВихрь Бюргерса (3D):")
t, r, z = 1.0, 1.0, 1.0
v_r = -kappa * r
v_phi = (Gamma/(2*math.pi*r)) * (1 - math.exp(-r**2/(4*nu*t)))
v_z = 2 * kappa * z
print(" |v| = %0.4f, div(v) = %0.6f" % (math.sqrt(v_r**2+v_phi**2+v_z**2), -2*kappa+2*kappa))

#
=====

```

```

# ШАГ 4: СТРУКТУРА mod 10
#
=====
print("\n" + "=" * 80)
print("ШАГ 4: СТРУКТУРА НЕЛИНЕЙНЫХ ВЗАИМОДЕЙСТВИЙ")
print("=" * 80)

count_s2s = 0
count_s2u = 0
for m1 in range(10):
    for m2 in range(10):
        m_sum = (m1 + m2) % 10
        if is_stable(m1) and is_stable(m2):
            if is_stable(m_sum):
                count_s2s += 1
            else:
                count_s2u += 1

print("Стабильные -> Стабильные: %d из 36" % count_s2s)
print("Стабильные -> Нестабильные: %d из 36 (%.1f%%)" % (count_s2u,
100*count_s2u/36))

#
=====
# ШАГ 5: ЗАТУХАНИЕ НЕСТАБИЛЬНЫХ МОД
#
=====
print("\n" + "=" * 80)
print("ШАГ 5: ЗАТУХАНИЕ НЕСТАБИЛЬНЫХ МОД")
print("=" * 80)

unstable = [1, 4, 8, 9]
for m in unstable:
    k = 2 * math.pi * m / n_tet
    t_decay = 1.0 / (nu * k**2)
    print("m=%d: t_decay = %.2f" % (m, t_decay))

#
=====
# ИТОГ
#
=====
print("\n" + "=" * 80)
print("ИТОГ")
print("=" * 80)

print("""
1. СР-фильтр — непрерывный оператор в  $H^s$ ,  $\|P\| = 1$ .

```

2. $[P, S] = 0$ — строго доказано.
 3. $v(t) = \exp(vt \nabla^2) v_0$ — факторизация для РН.
 4. Вихри Ламба-Озена и Бюргерса — частные случаи.
 5. Структура mod 10: 33% перекачки в нестабильные моды.
 6. Нестабильные моды затухают за конечное время.
- """)

```
print("=" * 80)
print("ВСЕ РЕЗУЛЬТАТЫ ВОСПРОИЗВОДИМЫ")
print("=" * 80)
```

Конец препринта