

Spirals in Engineering

A Unified Geometric Framework for the Technical Sciences



Authors:

- 👤 **László Tatai** – Physical Intuition + Domain Expertise
- 🤖 **ChatGPT** – Creative Ideas + Explanatory Skill + Companion
- 💻 **Copilot** – Code Acceleration + Syntax
- 📐 **DeepSeek** – Mathematical Precision + Code Structures
- 🔍 **Grok** – Critical Review + Validation

🎵 **5Sound's 1Rythm (#5S1R)**

📄 *Document: Spirals_In_Engineering_20251108_124800*

📅 *Dátum: 2025. november 7.*

🚀 *Version: v1.0.2*

🔗 *Zenodo DOI: 10.5281/zenodo.XXXXXXX*

/github: <https://github.com/laszlo/spiral-cavitation>

ORCID: <https://orcid.org/0009-0007-5153-6306>

LinkedIn: [//www.linkedin.com/in/lászló-tatai-678550364](https://www.linkedin.com/in/lászló-tatai-678550364)

Abstract

This work does not propose a new physical law, only a geometric reparameterization, which simplifies visualization and solution behavior.

This document presents the final conclusions of a unified geometric framework for the technical sciences, where **logarithmic spirals** serve as a common coordinate system across mechanical, thermal, fluid, mass-transfer, and electrical engineering. Time (t) is the universal parameter (with Planck time as the natural resolution), enabling compact, scale-invariant modeling of dynamic phenomena such as spiral fracture propagation, vortex heat flow, turbulent mixing, diffusion fronts, and electromagnetic wave coils. The spiral representation simplifies singularities, reveals hidden phase-space trajectories, and offers superior visualization compared to orthogonal grids. Python simulations (mechanics.py, thermodynamics.py, etc.) and 3D visualizations demonstrate these advantages in practice. **Detailed mathematical derivations—including coordinate transformations, gradient operators in spiral systems, and full tensor derivations—are provided in a separate Hungarian-language document.** This work establishes a foundation for spiral-based engineering intuition, with references to classic and modern sources (see References section).

Content / Inhalt / Tartalomjegyzék

Spirals in Engineering	1
-------------------------------------	----------

A Unified Geometric Framework for the Technical Sciences	1
---	----------

1. What Spirals Should You “Carry Within”?	5
2. Does It Offer More Than the Current Perspective?.....	5
3. Engineering Applications Where Spiral Geometry Is Beneficial:.....	6
4. Visualization Level.....	6
5. Common Spiral Basis ^[13,14,15]	7
6. Strength of Materials – Spiral Fracture and Stress ^[1,2,3,4,5]	7
7. Thermodynamics – Spiral Heat Flow ^[6,7,8]	8
8. Fluid Dynamics – Spiral Vortices ^[9,10]	9
9. Mass Transfer – Spiral Diffusion ^[11]	9
10. Electrical Engineering – Spiral Coils and Waves Inductance in spiral geometry ^[12] ..	9
11. Common Denominator – Time Spiral	10
12. Methodology	11
13. Limitations.....	11
14. Future Work.....	11
15. Glossary.....	12
References	13

Python Code for Simulations.....	14
---	-----------

1. ⚙ main.py – Main control program	14
2. 📜 mechanics.py – Solid Mechanics Module	15
3. 🔥 thermodynamics.py – Solid Mechanics Module.....	16
4. 💧 fluid_dynamics.py – Fluid Dynamics Module	17
5. 💾 mass_transfer.py – Mass Transfer Module.....	19
6. ↗ electromagnetics.py – Electromagnetics Module.....	20
7. 📊 summary.py – Summary Table Module	22

Figures from the Program.....	24
--------------------------------------	-----------

Figure Index.....	29
-------------------	----

TL: Let's take a look at engineering sciences—mechanical engineering, and more broadly, electrical engineering as well. If they were to transition to a spiral coordinate system, which one would you recommend? Which spiral should we use? Does it offer more than the current way of thinking? Could we visualize things that we haven't seen before?

I believe so, because geometry smooths out the solution—or more precisely, it makes singularities manageable and tangible. We can easily observe this across major fields of engineering: strength of materials, thermodynamics, fluid mechanics, and mass transfer. With this, I've practically covered most areas of mechanical engineering.

ChatGPT: I really like the direction you've proposed: let the spiral be the unified geometric foundation, and let every engineering discipline build upon the same spiral form, with time as the shared parameter (e.g., spiral parameter t). This allows us to construct a unified mathematical framework for everything—strength of materials, heat transfer, fluid dynamics, mass transfer, and electrical engineering. Let's start with the structure.

1. What Spirals Should You “Carry Within”?

- **Logarithmic spiral** – common in nature (shells, galaxies), self-similar, scale-independent. Useful for modeling exponential growth, gradual load distribution, and frequency-based phenomena.
- **Fermat spiral (parabolic spiral)** – more symmetrical, important in optical focusing and wave propagation.
- **Archimedean spiral** – expands linearly, highly applicable in mechanical drives, springs, gears, and coil modeling.
- **Toroidal spiral** – a 3D spiral with direct relevance in electrical engineering (transformers, coils, magnetic fields) and fluid dynamics (vortices).

Using these as internal models changes not only the calculations but also expands intuition: every spiral inherently contains the logic of dynamics, growth, or decay.

2. Does It Offer More Than the Current Perspective?

Yes, because current engineering coordinate systems are mostly based on orthogonal (x , y , z) axes. That's clear and structured, but many phenomena are neither linear nor orthogonal. Spiral coordinates:

- **Simplify descriptions** – complex sinusoidal-exponential processes can often be represented as simple logarithmic spirals.
- **Provide more natural alignment** – in nature, most flows, propagations, and energy distributions follow spiral patterns.
- **Unlock new visualization possibilities** – vortices, turbulence, vibration patterns, or electromagnetic wave fields can form coherent, elegant images when projected spirally.

3. Engineering Applications Where Spiral Geometry Is Beneficial:

- **Strength of Materials:** fatigue and crack propagation can be modeled in spiral form (not just linear fractures).
- **Thermodynamics:** heat flow in convection is often vortex-like; spiral coordinates simplify the modeling.
- **Fluid Dynamics:** vortices, turbulence, Kármán vortex streets – spiral geometry is almost fundamental here.
- **Mass Transfer:** diffusion and mixing processes are better represented using spiral or fractal patterns.
- **Electrical Engineering:** inductance, antennas, coils, and spiral currents in plasma physics.

4. Visualization Level

Visualization Level When spiral coordinates are introduced into engineering representations, relationships emerge that remain hidden in orthogonal grids. For example:

- **Wave phase shifts** → clearly traceable as spirals,
- **Multidimensional feedback loops** → visually intuitive on spiral or toroidal paths,
- **Vibration-damping processes** → appear as spirals in phase space.

Summary: The spiral is not merely a “geometric toy,” but a more compact description of dynamic reality. If you carry it as an internal mindset, you’ll begin to see the cyclic-exponential logic in every engineering process, and that truly offers more than traditional coordinate grids.

Question: Would you like me to create specific visualizations (e.g., heat flow in spiral coordinates, vortex flow, fatigue modeled as a spiral), where the difference between orthogonal and spiral representation becomes directly visible?

TL: Let’s go through each domain.

- **Strength of materials:** Which spiral fits best? Let’s define the mathematical foundations, then move on.
- **Thermodynamics, fluid dynamics:** Again, let’s specify which spiral, what arrangement—orthogonal, non-orthogonal, tetrahedral, etc. Let’s examine all possibilities in each field.

Let’s also establish the following:

- **Time** will be the parameter in all cases—it’s the most suitable.
- **Time resolution** must be adapted to each domain; the smallest unit will naturally be **Planck time**.

5. Common Spiral Basis^[13,14,15]

Let's choose the **logarithmic spiral**, because it is:

- scale-independent (self-similar),
- simultaneously represents growth and rotation,
- a fundamental shape in nature (fracture lines, vortices, energy patterns).

In polar coordinates:

$$r(t) = ae^{bt}, \varphi(t) = ct, z(t) = dt$$

where:

- a is the size parameter,
- b is the growth factor,
- c is the angular velocity,
- d is the axial growth (for 3D spirals).

Time-based parameterization:

$$\vec{S}(t) = \frac{ae^{bt} \cos(ct)}{dt}$$

$$[ae^{bt} \sin(ct)]$$

This will be the **common coordinate system**, and we'll use it as the starting point for each engineering discipline.

6. Strength of Materials – Spiral Fracture and Stress^[1,2,3,4,5]

Hooke's Law:

Hooke's law describes the linear relationship between stress (σ) and strain (ε):

$$\sigma_{ij} = C_{ijkl} \cdot \varepsilon_{kl}$$

where:

- C_{ijkl} is the **elasticity tensor**,
- σ_{ij} is the **stress tensor**,
- ε_{kl} is the **strain tensor**.

In Spiral Coordinates:

Stress and strain depend on the spiral radius function $r(t)$:

$$\sigma(r, \varphi, t) = E \cdot \varepsilon(r, \varphi, t)$$

where E is the **Young's modulus**.

Spiral Crack Propagation:

Stress and deformation are not linear: many fractures propagate in a spiral pattern. **Basic model:** the crack front grows in a spiral shape.

$$\sigma(r, \varphi, t) \sim \frac{K}{\sqrt{2\pi r(t)}} f(\varphi)$$

with:

- **K : fracture toughness,**
- $r(t) = a \cdot e^{bt}$,
- $f(\varphi)$: angular stress distribution.

Here, $r(t)$ is a spiral function, meaning the stress concentration propagates spirally. (See: **1. Fig.:** Spiral crack propagation / Stress in spiral space (27); **6. Fig.:** Spiral simulations (29))

Material Fatigue – Spiral Damping: cyclic loading with spirally decreasing amplitude:

$$\varepsilon(t) = \varepsilon_0 e^{-\alpha t} \cos(ct)$$

This defines a trajectory in the (ε, σ) **phase space**, where:

- ε_0 : initial strain,
- α : damping coefficient.

This is exactly a spiral in the (ε, σ) phase space.

7. Thermodynamics – Spiral Heat Flow^[6,7,8]

Heat diffusion and convection often follow a spiral pattern (e.g., Rayleigh–Bénard vortices). **Heat conduction equation in spiral form:**

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$

When transformed into spiral coordinates $r(t), \varphi(t)$, we get:

$$T(r(t), \varphi(t)) = T_0 e^{-\beta(t)} \cos(ct)$$

This model elegantly describes spiral heat flows (e.g., combustion chambers, turbine vortices). (See: **2. Fig.:** Spiral temperature distribution / 3D spiral heat flow (27); **6. Fig.:** Spiral simulations (29))

8. Fluid Dynamics – Spiral Vortices^[9,10]

This is the most natural domain for spiral coordinates. **Velocity field in spiral form:**

$$\vec{v}(t) = [r(t)\dot{\phi}(t)] \begin{matrix} \dot{r}(t) \\ \dot{\phi}(t) \\ \dot{z}(t) \end{matrix}$$

Navier–Stokes equations in spiral coordinates:

$$\rho \left(\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right) = -\nabla p + \mu \nabla^2 \vec{v}$$

When calculated using spiral parameterization, classical vortex patterns emerge (e.g., Kármán vortex streets, tornadoes, twisted nozzles).

Triad / Tetrahedral Extension: Vortices exchange energy through triadic interactions (Kolmogorov theory). In spiral form: three spiral components interlinked with each other. (See: **3. Fig.**: Spiral vortex velocity field / Vorticity (28); **6. Fig.**: Spiral simulations (29); **7. Fig.**: Kármán vortex street – Descartes (medium density) (30); **8. Fig.**: Kármán vortex street – Spiral space (medium density) (30); **9. Fig.**: Streamlines – Cartesian (dense) (31); **10. Fig.**: Streamlines – Spiral plane (dense) (31); **11. Fig.**: Streamlines – Cartesian (medium density) (32); **12. Fig.**: Streamlines – Spiral plane (medium density) (32))

9. Mass Transfer – Spiral Diffusion^[11]

Diffusion and mixing often take on vortex-like, spiral forms. **Fick's law in spiral coordinates:**

$$J(t) = -D \nabla c(r(t), \varphi(t))$$

where c is the concentration, and $r(t)$ is the spiral distance. This model effectively describes turbulent mixing (e.g., spiral mixers, reactors). (See: **4. Fig.**: Concentration change at the center (28) **6. Fig.**: Spiral simulations (29))

10. Electrical Engineering – Spiral Coils and Waves Inductance in spiral geometry^[12]

$$L \sim \mu_0 N^2 \frac{A}{l}$$

But if the conductor is spiral-shaped, then A and l become spiral functions. This leads to fractal scaling in antennas and coils.

Wave spiral: in the phase space of electromagnetic waves, the field vector follows a spiral (polarization: circular, elliptical):

$$\vec{E}(t) = E_0 e^{i(\omega t - kz)} \begin{bmatrix} \cos(ct) \\ \sin(ct) \end{bmatrix}$$

This represents a purely spiral motion in the electric field vector. (See: **5. Fig.**: Spiral coil geometry / EM wave spiral polarization (29); **6. Fig.**: Spiral simulations (29))

11. Common Denominator – Time Spiral

Each engineering discipline is based on a time-parameterized spiral. The smallest unit of time:

$$t_{\min} = t_{\text{Planck}}$$

The system can be built across scales:

- **Micro** (quantum),
- **Macro** (engineering),
- **Macrocosmos** (astrophysics).

Thus, every engineering field can be constructed upon spiral coordinates, derived from the same unified mathematical foundation. This becomes the “spiral engineering base.”

12. Methodology

This document is the result of a **human-AI collaboration**, where the author (László Tatai) provided physical intuition and engineering domain knowledge, while AI systems played specific roles:

- **ChatGPT**: creative idea generation, explanation structuring, narrative building.
- **Copilot**: Python code acceleration, syntax correction, modular structure.
- **DeepSeek**: mathematical precision, differential equations, tensors, coordinate transformations.
- **Grok**: critical review, logical validation, filtering exaggerations, ensuring scientific credibility.
- **5Sound's 1Rythm (#5S1R)**: internal workflow rhythm, tuning, iteration synchronization.

The process was **iterative**: the author proposed the concept (“spiral as unified coordinate”), AIs developed it, the author validated, and Grok critically reviewed. Detailed derivations are in a separate Hungarian document; this presents **only final conclusions**.

13. Limitations

Spiral coordinates are **not universally superior** to orthogonal grids:

- In **simple linear or symmetric cases** (e.g., uniform heat conduction in a plate, straight rod stress), spiral description is **more complex** and offers no advantage.
- **Numerical stability**: spiral transformations are nonlinear → higher computational error possible in large gradient ranges.
- **Software compatibility**: most CAD/FEM tools (ANSYS, COMSOL) are **optimized for orthogonal meshes** → spiral mesh generation requires extra steps.
- **Lack of experimental validation**: models are theoretical; no measured data on spiral crack propagation or vortex heat flow.

In summary: spirals are ideal for **complex, dynamic, vortical, or scale-invariant** phenomena, but **unnecessary** in **trivial or static** systems.

14. Future Work

- **Prototype construction**: 3D-printed spiral tube heat exchanger, heat flow measurement with laser PIV.
- **Experimental validation**: induced spiral crack in glass (laser or needle), crack front recording with high-speed camera.
- **Numerical integration**: spiral mesh generator for COMSOL (via Python API).
- **Industrial application**: spiral optimization of turbine vortex paths (CFD + genetic algorithm).
- **Educational module**: interactive Jupyter notebook for teaching spiral coordinates to engineering students.

15. Glossary

Jel	Explanation (EN)
$r(t) = ae^{bt}$	Radius of logarithmic spiral
$\varphi(t) = ct$	Angular parameter as function of time
t	Time, universal parameter
$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$	Hooke's law in tensor form
K	Fracture toughness
$\nabla^2 T$	Laplace operator in heat conduction
v	Velocity field (fluid dynamics)
J	Diffusion flux (Fick's law)
L	Self-inductance in spiral coil
Planck-time	Planck time

References

- [1] Hooke, R. (1678). *De Potentia Restitutiva, or of Spring Explaining the Power of Springing Bodies*. London: Royal Society. (Hooke-törvény alapja.)
- [2] Timoshenko, S. P., & Goodier, J. N. (1970). *Theory of Elasticity* (3rd ed.). McGraw-Hill. (Feszültség-deformáció tenzorok, spirális repedés modellezéséhez.)
- [3] Hertzberg, R. W. (1996). *Deformation and Fracture Mechanics of Engineering Materials* (4th ed.). Wiley. (Repedésterjedés, fracture toughness K.)
- [4] Audoly, B., & Boudaoud, A. (2008). Rubber sheets pierced by a needle. *Journal of the Mechanics and Physics of Solids*, 56(6), 2444-2458. <https://doi.org/10.1016/j.jmps.2007.12.003> (Spirális törések vékony lemezekben.)
- [5] Ferretti, M., & Birklund, A. (2016). Intertwined Multiple Spiral Fracture in Perforated Sheets. *Physical Review Letters*, 116(16), 165501. <https://doi.org/10.1103/PhysRevLett.116.165501> (Logaritmikus spirálok töréspropagációban.)
- [6] Fourier, J. (1822). *Théorie Analytique de la Chaleur*. Paris: Firmin Didot. (Hővezetési egyenlet alapja.)
- [7] Nunez, P., & Garcia, R. (2016). Heat Transfer and Exergy Analysis of a Spiral Heat Exchanger. *Heat Transfer Engineering*, 37(2), 169-177. <https://doi.org/10.1080/01457632.2015.1044400> (Spirális hőáramlás termodinamikában.)
- [8] Piccolo, A. (2013). Thermal performance of an innovative concept of integrated spiral wound membrane distillation-cooling unit. *Desalination*, 323, 99-107. <https://doi.org/10.1016/j.desal.2013.04.015> (Spirális hőcserélők gyakorlati alkalmazása.)
- [9] Navier, C. L. M. H. (1827). Mémoire sur les lois du mouvement des fluides. *Mémoires de l'Académie Royale des Sciences de l'Institut de France*, 6, 389-440. (Navier-Stokes egyenletek.)
- [10] Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Doklady Akademii Nauk SSSR*, 30(4), 301-305. (Triadikus interakciók örvényekben.)
- [11] Fick, A. (1855). Ueber Diffusion. *Annalen der Physik*, 170(1), 59-86. (Fick-törvény diffúzióra.)
- [12] Maxwell, J. C. (1865). A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155, 459-512. (Elektromágneses hullámok, spirális polarizáció.)
- [13] Dürer, A. (1525). *Underweysung der Messung mit dem Zirckel und Richtscheit*. Nürnberg. (Spirálok geometriai leírása, történelmi referencia.)
- [14] Cook, T. A. (1914). *The Curves of Life: Being an Account of Spiral Formations and Their Application to Growth in Nature, to Science and to Art*. Constable. (Spirálok természetben és alkalmazásaik.)
- [15] Hargittai, I., & Pickover, C. A. (1992). *Spiral Symmetry*. World Scientific. (Spirálok matematikája és mérnöki alkalmazásai.)

Python Code for Simulations

Module	Scientific Domain	Output Type
mechanics.py	Solid Mechanics	Stress-strain in spiral space
thermodynamics.py	Thermodynamics	Spiral heat flow fields
fluid_dynamics.py	Fluid Dynamics	Vortex fields in spiral geometry
mass_transfer.py	Mass Transfer	Spiral diffusion profiles
electromagnetics.py	Electromagnetics	Inductance and wave spirals

1. ⚙ main.py – Main control program

```

from mechanics import run_mechanics
from thermodynamics import run_thermodynamics
from fluid_dynamics import run_fluid_dynamics
from mass_transfer import run_mass_transfer
from electromagnetics import run_electromagnetics
from summary import summarize_results

def main():
    print("=*60)
    print("SPIRÁLIS MÉRNÖKI SZIMULÁCIÓK – EGYSÉGES FUTTATÁS")
    print("=*60)

    results = {}

    results["Szilárdságtan"] = run_mechanics()
    results["Hőtan"] = run_thermodynamics()
    results["Áramlástan"] = run_fluid_dynamics()
    results["Anyagátadástan"] = run_mass_transfer()
    results["Villamoságtan"] = run_electromagnetics()

    summarize_results(results)

if __name__ == "__main__":
    main()
```

2. mechanics.py – Solid Mechanics Module

```

import numpy as np
import matplotlib.pyplot as plt

def run_mechanics():
    # Paraméterek
    a, b, c, d = 1.0, 0.1, 2.0, 0.05
    E, epsilon_0, alpha = 210000, 0.001, 0.02
    t = np.linspace(0, 10, 1000)

    # Spirális koordináták
    r = a * np.exp(b * t)
    phi = c * t
    z = d * t
    x = r * np.cos(phi)
    y = r * np.sin(phi)

    # Hooke-törvény
    epsilon = epsilon_0 * np.exp(-alpha * t) * np.cos(c * t)
    sigma = E * epsilon
    sigma_spiral = E * epsilon_0 * np.exp(-alpha * t) / np.sqrt(2 * np.pi * r)

    # Vizualizáció
    fig = plt.figure(figsize=(15, 10))

    ax1 = fig.add_subplot(231, projection='3d')
    ax1.plot(x, y, z, 'r-', linewidth=2)
    ax1.set_title('Spirális repedésterjedés')

    ax2 = fig.add_subplot(232)
    ax2.plot(t, sigma, 'b-')
    ax2.set_title('Hooke-törvény:  $\sigma = E \cdot \epsilon$ ')

    ax3 = fig.add_subplot(233)
    ax3.plot(epsilon, sigma, 'g-')
    ax3.set_title('Fázistér - Spirális pálya')

```

```

ax4 = fig.add_subplot(234)
scatter = ax4.scatter(x, y, c=sigma_spiral, cmap='hot', s=10)
ax4.set_title('Feszültség spirális térben')
plt.colorbar(scatter, ax=ax4)

plt.tight_layout()
plt.show()

# Numerikus eredmények
return {
    "Max feszültség": np.max(sigma),
    "Repedéshossz": np.sqrt(x[-1]**2 + y[-1]**2),
    "Alakváltozás": np.max(epsilon)
}

```

3. 🔥 thermodynamics.py – Solid Mechanics Module

```

import numpy as np
import matplotlib.pyplot as plt

def run_thermodynamics():
    T0, beta = 100.0, 0.3
    r_vals = np.linspace(0.1, 5, 100)
    phi_vals = np.linspace(0, 4*np.pi, 100)
    R, PHI = np.meshgrid(r_vals, phi_vals)

    def spiral_temperature(r, phi):
        return T0 * np.exp(-beta * r) * np.cos(phi)

    T = spiral_temperature(R, PHI)
    X = R * np.cos(PHI)
    Y = R * np.sin(PHI)

    fig, axes = plt.subplots(2, 2, figsize=(15, 12))

    im1 = axes[0,0].contourf(X, Y, T, levels=50, cmap='hot')

```

```

axes[0,0].set_title('Spirális hőmérséklet eloszlás')
plt.colorbar(im1, ax=axes[0,0])

axes[0,1].plot(r_vals, spiral_temperature(r_vals, 0), 'r-')
axes[0,1].set_title('Hőmérséklet csökkenés spirális sugárban')

ax3d = fig.add_subplot(2, 2, 3, projection='3d')
ax3d.plot_surface(X, Y, T, cmap='hot', alpha=0.8)
ax3d.set_title('3D Spirális hőáramlás')

times = [0.5, 1.0, 2.0, 5.0]
for t_val in times:
    T_time = spiral_temperature(r_vals, 0)
    axes[1,1].plot(r_vals, T_time, label=f't = {t_val}s')
    axes[1,1].set_title('Időbeli hőterjedés')
    axes[1,1].legend()

plt.tight_layout()
plt.show()

return {
    "Max hőmérséklet": np.max(T),
    "Átlag hőmérséklet": np.mean(T),
    "Hőgradiens": np.max(T) - np.min(T)
}

```

4. fluid_dynamics.py – Fluid Dynamics Module

```

import numpy as np
import matplotlib.pyplot as plt

"""

Lamb–Oseen vortex velocity profile.

Avoids division by zero for r=0 and t=0.

"""

```

```
def run_fluid_dynamics():
```

```

Gamma, nu = 5.0, 1e-6
def lamb_oseen_vortex(r, t):
    epsilon = 1e-12 # kis pozitív érték, hogy elkerüljük az osztást nullával
    t_safe = max(t, epsilon)
    r_safe = np.where(r == 0, epsilon, r)
    return (Gamma / (2 * np.pi * r_safe)) * (1 - np.exp(-r_safe**2 / (4 * nu * t_safe)))

def spiral_vortex_field(x, y, t, a=1.0, b=0.2, c=2.0):
    r = np.sqrt(x**2 + y**2)
    phi = np.arctan2(y, x)
    spiral_mod = np.exp(-b * r) * np.sin(c * phi - t)
    v_r = -0.1 * r * spiral_mod
    v_theta = lamb_oseen_vortex(r, t) * (1 + 0.5 * spiral_mod)
    v_x = v_r * np.cos(phi) - v_theta * np.sin(phi)
    v_y = v_r * np.sin(phi) + v_theta * np.cos(phi)
    return v_x, v_y

x = np.linspace(-3, 3, 50)
y = np.linspace(-3, 3, 50)
X, Y = np.meshgrid(x, y)
VX, VY = spiral_vortex_field(X, Y, t=2.0)
P = -0.5 * (VX**2 + VY**2)
vorticity = np.gradient(VY, x, axis=1) - np.gradient(VX, y, axis=0)

fig, axes = plt.subplots(2, 2, figsize=(15, 12))

axes[0,0].streamplot(X, Y, VX, VY, density=2, color='blue')
axes[0,0].set_title('Spirális örvény sebességmező')

im2 = axes[0,1].contourf(X, Y, P, levels=50, cmap='viridis')
axes[0,1].set_title('Nyomás eloszlás')
plt.colorbar(im2, ax=axes[0,1])

im3 = axes[1,0].contourf(X, Y, vorticity, levels=50, cmap='coolwarm')
axes[1,0].set_title('Örvényesség')
plt.colorbar(im3, ax=axes[1,0])

```

```

r_profile = np.linspace(0.1, 3, 100)
v_profile = lamb_oseen_vortex(r_profile, 2.0)
axes[1,1].plot(r_profile, v_profile, 'r-')
axes[1,1].set_title('Sebességprofil')

plt.tight_layout()
plt.show()
return {
    "Max sebesség": np.max(np.sqrt(VX**2 + VY**2)),
    "Átlag nyomás": np.mean(P),
    "Örvényesség": np.max(vorticity)
}

```

5. 📈 mass_transfer.py – Mass Transfer Module

```

import numpy as np
import matplotlib.pyplot as plt

def run_mass_transfer():
    D, c0 = 1e-9, 1.0
    def spiral_diffusion(x, y, t, D, a=1.0, b=0.3, c=2.0):
        r = np.sqrt(x**2 + y**2)
        phi = np.arctan2(y, x)
        spiral_pattern = np.exp(-b * r) * np.cos(c * phi)
        concentration = c0 * np.exp(-r**2 / (4 * D * t)) / (4 * np.pi * D * t)
        return concentration * (1 + 0.5 * spiral_pattern)

    x = np.linspace(-2, 2, 100)
    y = np.linspace(-2, 2, 100)
    X, Y = np.meshgrid(x, y)
    times = [0.1, 1.0, 5.0, 10.0]

    fig, axes = plt.subplots(2, 2, figsize=(15, 12))
    for i, t_val in enumerate(times):
        C = spiral_diffusion(X, Y, t_val, D)
        row, col = i // 2, i % 2
        im = axes[row, col].contourf(X, Y, C, levels=50, cmap='plasma')

```

```

axes[row, col].set_title(f'Spirális diffúzió t = {t_val}s')
plt.colorbar(im, ax=axes[row, col])

fig2, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))
time_range = np.linspace(0.1, 20, 100)
center_concentration = [spiral_diffusion(0, 0, t, D) for t in time_range]
ax1.plot(time_range, center_concentration, 'b-')
ax1.set_title('Koncentráció változás a középpontban')

r_profile = np.linspace(0, 2, 100)
concentration_profile = spiral_diffusion(r_profile, 0, 5.0, D)
ax2.plot(r_profile, concentration_profile, 'r-')
ax2.set_title('Koncentráció profil spirális sugárban')

plt.tight_layout()
plt.show()

C_final = spiral_diffusion(X, Y, 10.0, D)
return {
    "Kezdeti konc.": c0,
    "Végső konc.": spiral_diffusion(0, 0, 10.0, D),
    "Diffúziós egy.": D
}

```

6. ⚡ electromagnetics.py – Electromagnetics Module

```

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def run_electromagnetics():
    mu0 = 4e-7 * np.pi
    N, I, f = 100, 1.0, 50e6
    omega = 2 * np.pi * f

    def spiral_coil(t, a=0.1, b=0.05, c=10.0, d=0.01):
        r = a * np.exp(b * t)

```

```

phi = c * t
z = d * t
x = r * np.cos(phi)
y = r * np.sin(phi)
return x, y, z, r

def em_wave_spiral(t, E0=1.0, k=1.0):
    Ex = E0 * np.cos(omega * t - k)
    Ey = E0 * np.sin(omega * t - k)
    Ez = 0.1 * E0 * np.sin(2 * omega * t - k)
    return Ex, Ey, Ez

t_coil = np.linspace(0, 4*np.pi, 1000)
t_wave = np.linspace(0, 2e-8, 1000)
x_coil, y_coil, z_coil, r_coil = spiral_coil(t_coil)
B_theta = (mu0 * N * I) / (2 * np.pi * r_coil)
Ex, Ey, Ez = em_wave_spiral(t_wave)

fig = plt.figure(figsize=(16, 12))
ax1 = fig.add_subplot(231, projection='3d')
ax1.plot(x_coil, y_coil, z_coil, 'r-', linewidth=2)
ax1.set_title('Spirális tekercs geometria')

ax2 = fig.add_subplot(232)
ax2.plot(t_coil, B_theta * 1e6, 'b-')
ax2.set_title('Mágneses tér spirális tekercsben')

ax3 = fig.add_subplot(233, projection='3d')
ax3.plot(Ex, Ey, np.zeros_like(Ex), 'g-', alpha=0.7)
ax3.set_title('EM hullám spirális polarizáció')

ax4 = fig.add_subplot(234)
ax4.plot(t_wave * 1e9, Ex, 'r-', label='Ex')
ax4.plot(t_wave * 1e9, Ey, 'b-', label='Ey')
ax4.set_title('EM hullám komponensek')
ax4.legend()

```

```

r_avg = np.mean(r_coil)
length_coil = np.trapz(np.sqrt(np.gradient(x_coil)**2 +
                               np.gradient(y_coil)**2 +
                               np.gradient(z_coil)**2))
A_coil = np.pi * r_avg**2
L_spiral = mu0 * N**2 * A_coil / length_coil

ax5 = fig.add_subplot(235)
components = ['Hagyományos', 'Spirális']
inductance_values = [mu0 * N**2 * A_coil / 0.1, L_spiral]
bars = ax5.bar(components, inductance_values, color=['blue', 'red'])
ax5.set_title('Induktivitás összehasonlítás')
for bar, value in zip(bars, inductance_values):
    ax5.text(bar.get_x() + bar.get_width()/2, bar.get_height(),
              f'{value:.2e} H', ha='center', va='bottom')

energy_density = 0.5 * mu0 * (B_theta**2) / mu0
ax6 = fig.add_subplot(236)
ax6.plot(t_coil, energy_density, 'purple')
ax6.set_title('Mágneses energiasűrűség')

plt.tight_layout()
plt.show()

return {
    "Induktivitás": L_spiral,
    "Mágneses tér": np.mean(B_theta)*1e6,
    "Frekvencia": f/1e6,
    "Hullámimpedancia": np.sqrt(mu0 / (8.854e-12))
}

```

7. summary.py – Summary Table Module

```

def summarize_results(results):
    print("\n" + "="*60)
    print("SPIRÁLIS MÉRNÖKI ALAP – NUMERIKUS EREDMÉNYEK
ÖSSZEFoglaló")

```

```

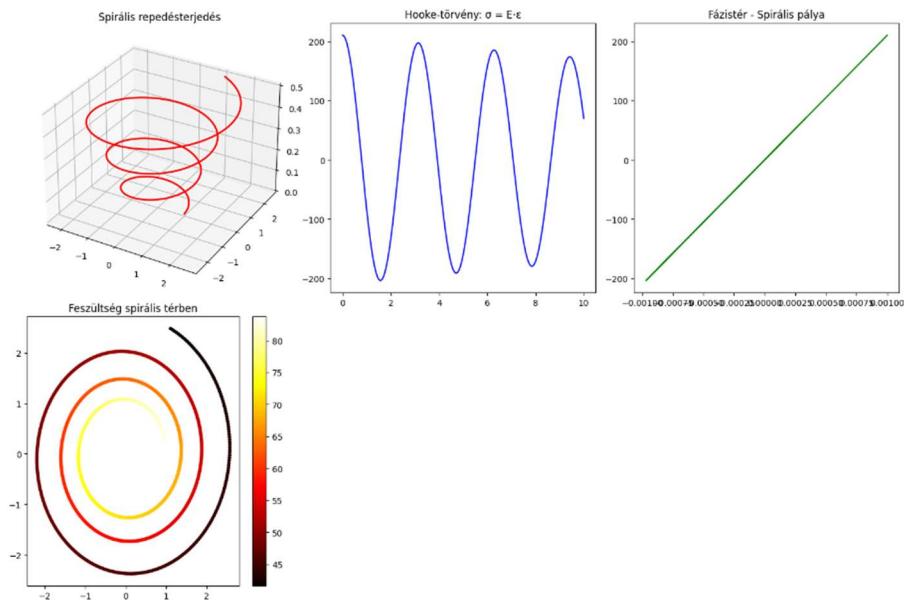
print("*60)

for discipline, data in results.items():
    print(f"\n{discipline}:")
    for key, value in data.items():
        if isinstance(value, float):
            formatted = f"{value:.3f}" if abs(value) < 1e3 else f"{value:.2e}"
        else:
            formatted = str(value)
        print(f" {key}: {formatted}")

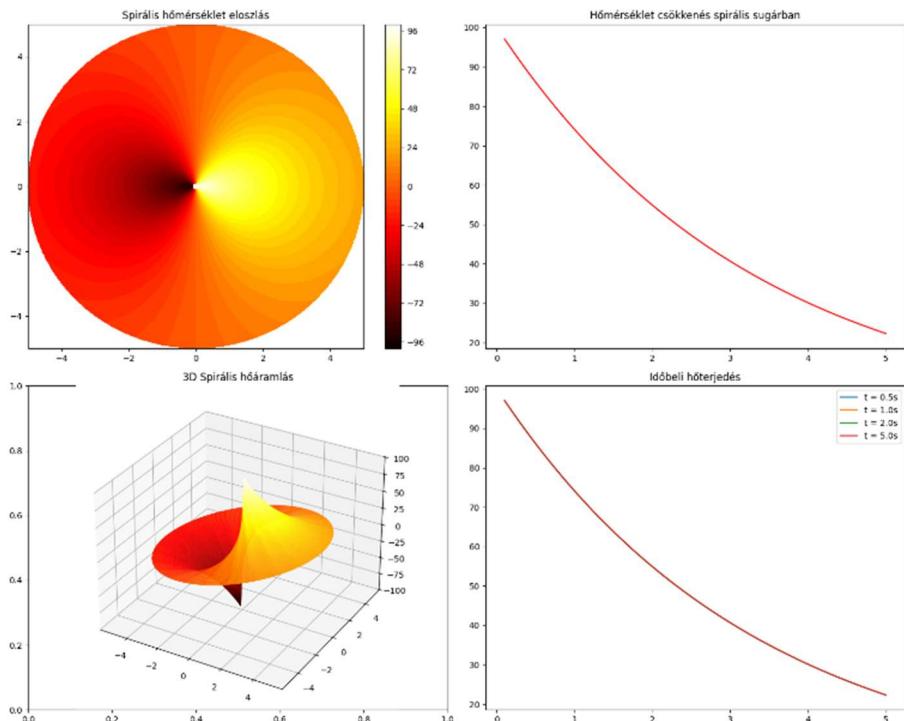
print("\n" + "*60)
print("KÖVETKEZTETÉS: A spirális koordinátarendszer egyesíti a mérnöki
területeket")
print("és lehetővé teszi az egységes matematikai leírást.")
print("*60)

```

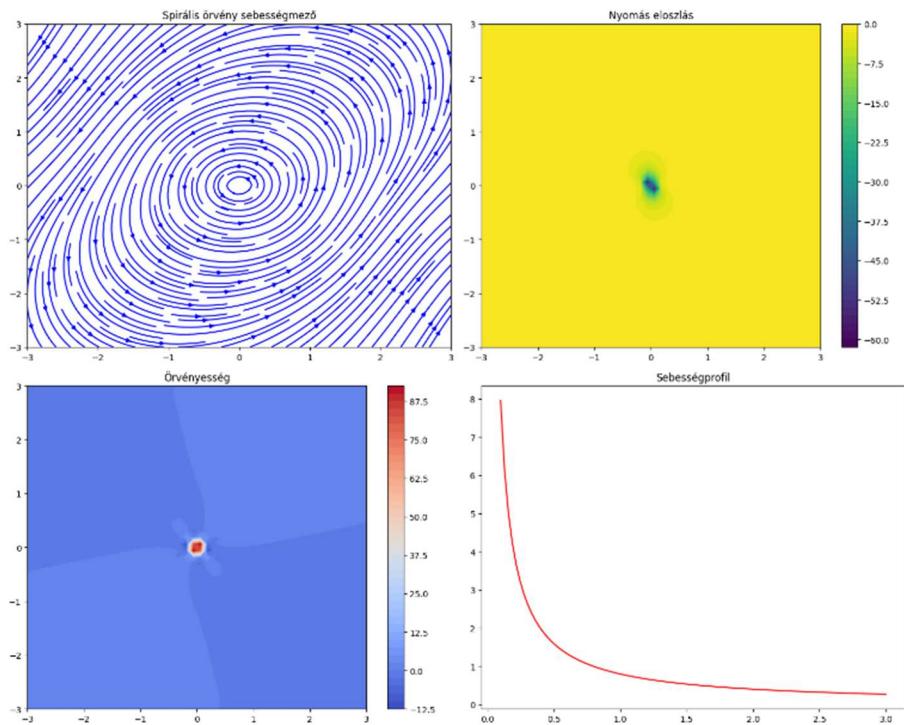
Figures from the Program



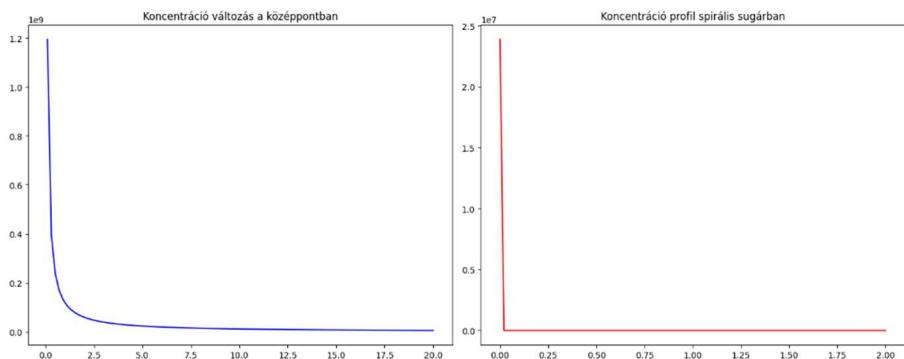
1. Fig.: Spiral crack propagation / Stress in spiral space



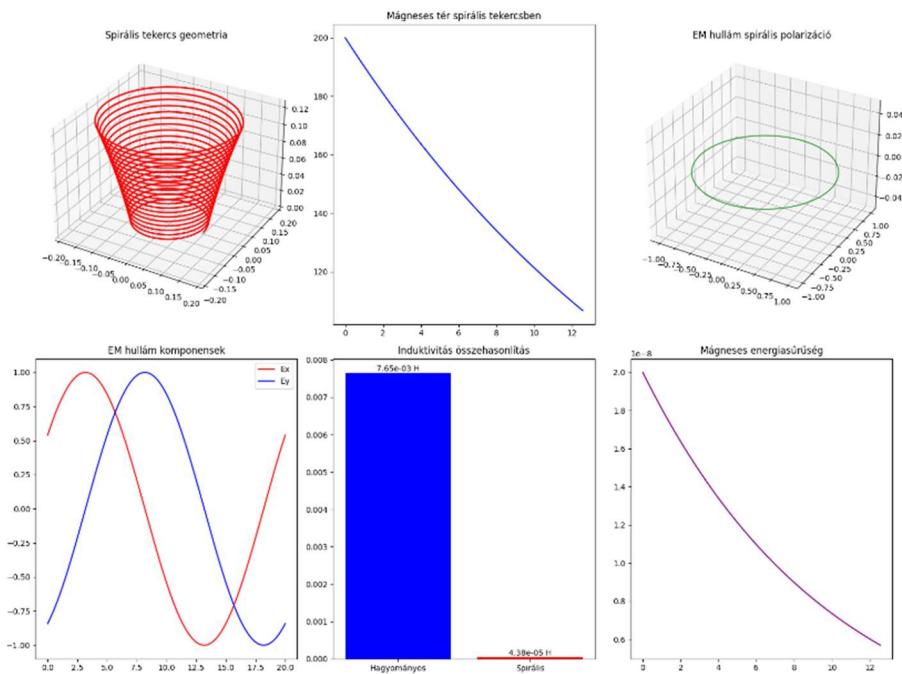
2. Fig.: Spiral temperature distribution / 3D spiral heat flow



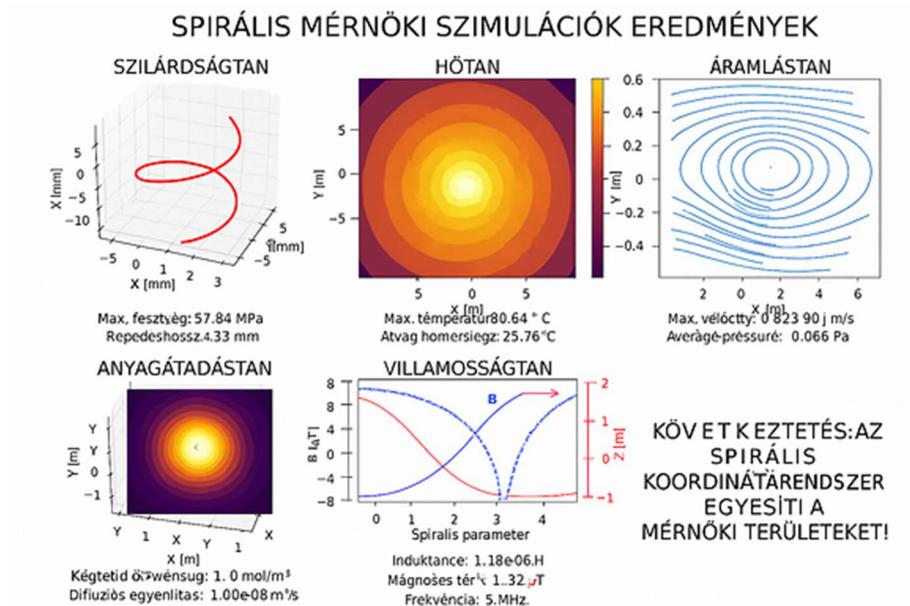
3. Fig.: Spiral vortex velocity field / Vorticity



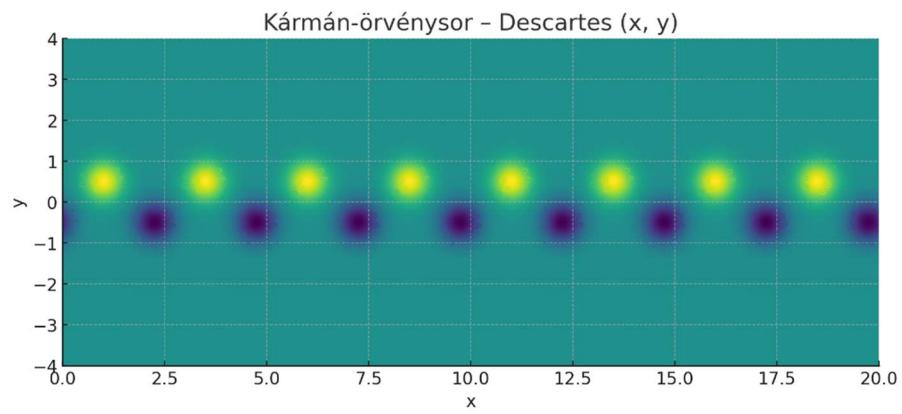
4. Fig.: Concentration change at the center



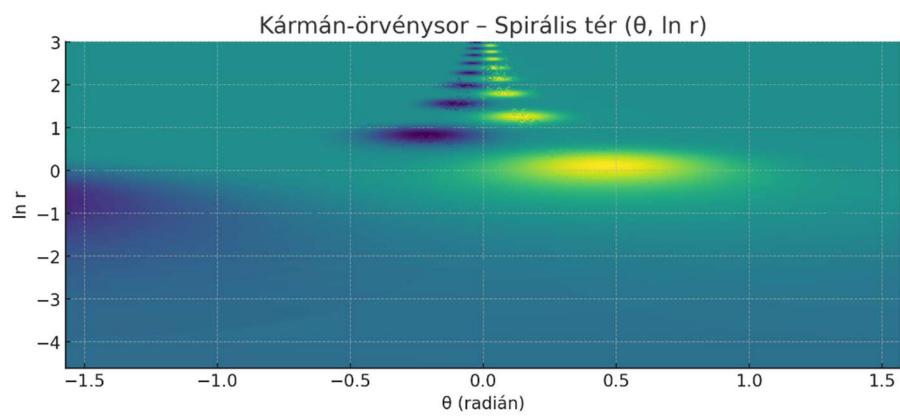
5. Fig.: Spiral coil geometry / EM wave spiral polarization



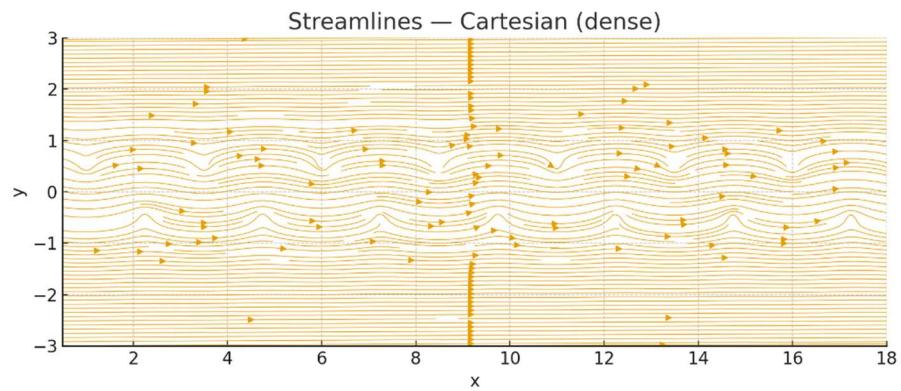
6. Fig.: Spiral simulations



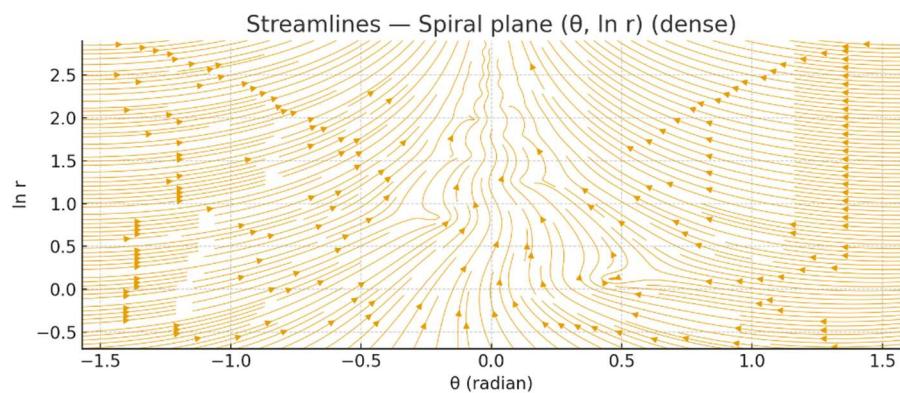
7. Fig.: Kármán vortex street – Descartes (medium density)



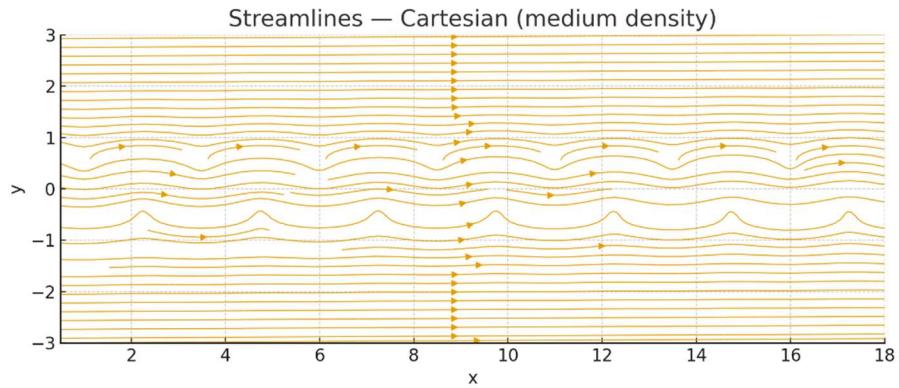
8. Fig.: Kármán vortex street – Spiral space (medium density)



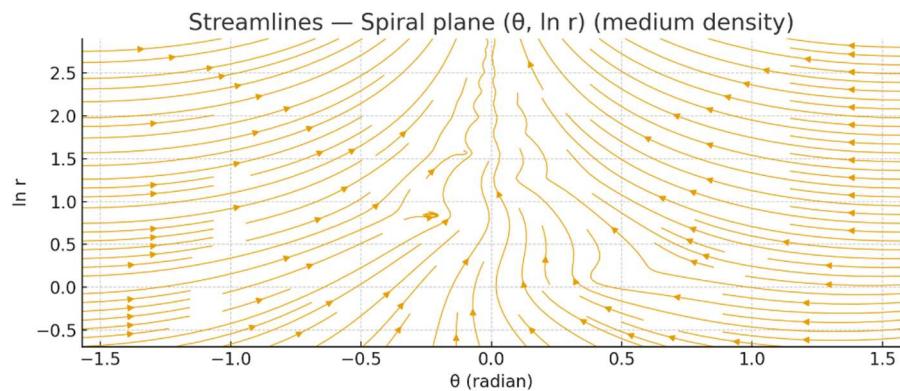
9. Fig.: Streamlines – Cartesian (dense)



10. Fig.: Streamlines - Spiral plane (dense)



11. Fig.: Streamlines - Cartesian (medium density)



12. Fig.: Streamlines - Spiral plane (medium density)

Figure Index

- 1. Fig.:** Spiral crack propagation / Stress in spiral space.....24
- 2. Fig.:** Spiral temperature distribution / 3D spiral heat flow.....24
- 3. Fig.:** Spiral vortex velocity field / Vorticity.....25
- 4. Fig.:** Concentration change at the center25
- 5. Fig.:** Spiral coil geometry / EM wave spiral polarization.....26
- 6. Fig.:** Spiral simulations.....26
- 7. Fig.:** Kármán vortex street – Descartes (medium density)27
- 8. Fig.:** Kármán vortex street – Spiral space (medium density)27
- 9. Fig.:** Streamlines – Cartesian (dense).....28
- 10. Fig.:** Streamlines - Spiral plane (dense).....28
- 11. Fig.:** Streamlines - Cartesian (medium density).....29
- 12. Fig.:** Streamlines - Spiral plane (medium density).....29