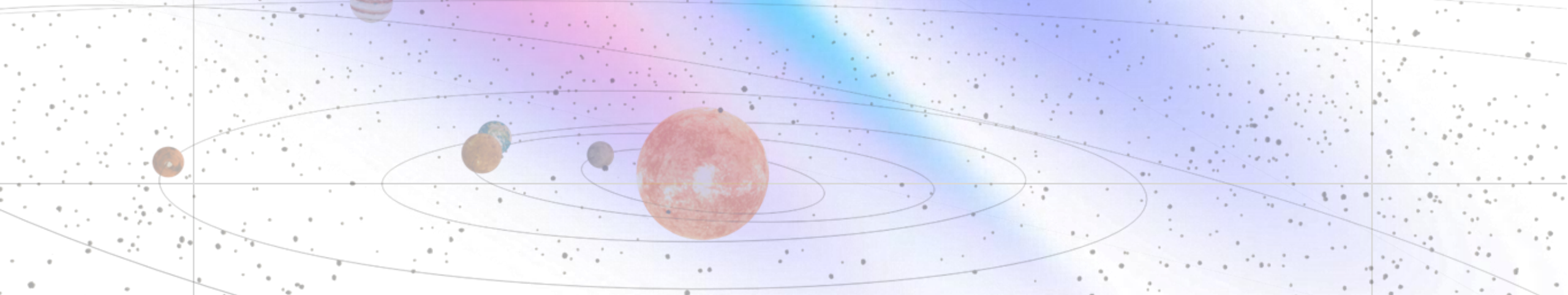


Die Zukunft unseres Sonnensystems

Wo liegen die Grenzen der Vorhersagbarkeit?

Referent: Leander Riefel / Betreuende Lehrkraft: Herr Weber / Hauptfach: Physik / Nebenfach: Informatik

Gliederung

- Grundlagen
- Chaos
- Simulation
- Vorhersagen
- Weitere Zukunftsaussichten
- Fazit

Grundlagen
Unser Sonnensystem

- 8 Planeten
- 5 Zwergplaneten
- 300+ Monde
- ca. 4000 Kometen
- ca. 1,4 Millionen Asteroiden

Grundlagen
Newtonsches Gravitationsgesetz

$$F = G\frac{m_1 m_2}{r^2}$$

$$\vec{F} = G\frac{m_1 m_2}{|\vec{r}|^2}\hat{r} \quad \rightarrow \quad \vec{F} = G\frac{m_1 m_2}{|\vec{r}|^3}\vec{r}$$

$$\overrightarrow{F_{12}} = -\overrightarrow{F_{21}}$$

Grundlagen
Keplersche Gesetze

1. Planeten bewegen sich auf Ellipsen (die Sonne in einem Brennpunkt).
2. Fahrstrahl überstreicht in gleichen Zeiten gleiche Flächen.
3. $P^2 \propto a^3$ – Bahnperiode P und große Halbachse a

Grundlagen
2-Körper-Problem

- Beispiel: Erde-Mond / Sonne-Erde
- Kann Analytisch gelöst werden:

$$M = E - e * \sin(E)$$

$$\tan\left(\frac{T}{2}\right) = \sqrt{\frac{1+e}{1-e}} * \tan\left(\frac{E}{2}\right)$$

$$r = \frac{a(1-e^2)}{1 + e * \cos(T)}$$

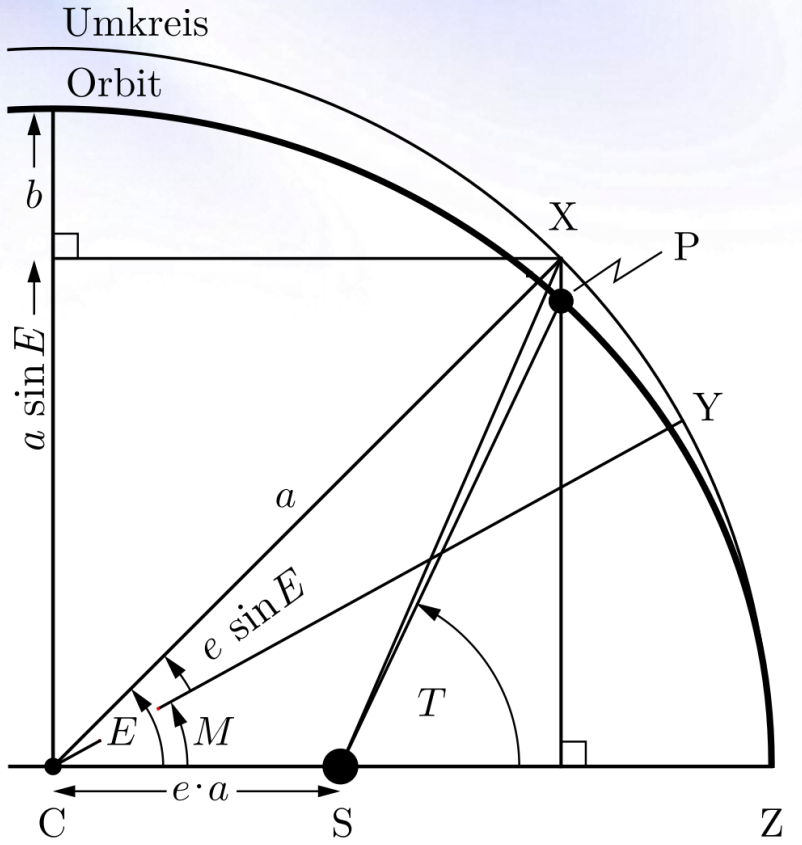


Abb. 1: Zusammenhänge der Kepler-Gleichung

Grundlagen
Analytisches vs. Numerisches Lösen

Analytisches Lösen

- Geschlossene Formel
- Exakte Lösung
- Niedriger Rechenaufwand
- Stabile Systeme » keine wachsenden Fehler

Numerisches Lösen

- Schrittweise Integration
- Näherung
- Hoher Rechenaufwand
- Instabil » Fehler wachsen mit jedem Schritt

Grundlagen
2-Körper-Problem

Hier Simulation eines 2-Körper-Systems

Grundlagen
n-Körper-Problem

- Muss für "$n_{Körper} > 2$" numerisch gelöst werden
  - Ausnahmen wurden von unteranderem Euler oder Lagrange gefunden
- Beispiel: Sonne-Erde-Mond
- Allgemein nicht-periodisch und chaotisch
- Sehr hoher Rechenaufwand

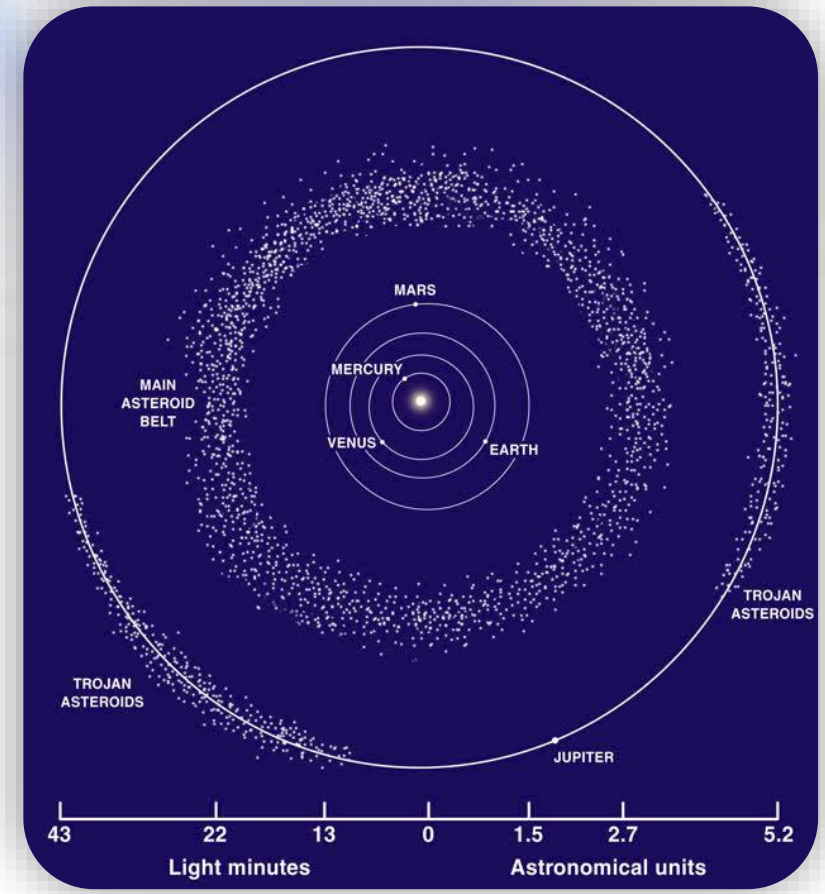


Abb. 2: Asteroidengürtel & Positionen

Grundlagen
n-Körper-Problem

$$\overrightarrow{F(t)} = G\frac{m_i m_j}{\left|\overrightarrow{r_{ji}}\right|^3}\overrightarrow{r_{ji}}$$

$$\vec{F} = m * \vec{a}$$

Grundlagen
n-Körper-Problem

$$\vec{F} = m * \vec{a} \quad \Big| \quad \vec{F} = m * \vec{a}, \qquad * \frac{1}{m_i}$$

$$\overrightarrow{a_i} = G \frac{m_j}{\left|\overrightarrow{r_{ji}}\right|^3} \overrightarrow{r_{ji}}$$

Grundlagen
n-Körper-Problem

$$\overrightarrow{F(t)} = G\frac{m_i m_j}{\left|\overrightarrow{r_{ji}}\right|^3}\overrightarrow{r_{ji}} \quad \Big| \quad \vec{F} = m * \vec{a}, \qquad * \frac{1}{m_i}$$

$$\overrightarrow{a_i} = G\frac{m_j}{\left|\overrightarrow{r_{ji}}\right|^3}\overrightarrow{r_{ji}}$$

$$\Downarrow$$

$$\overrightarrow{a_i(t)} = G\sum_{i \neq j} m_i \frac{r_{ji}(t)}{\left\|r_{ji}(t)\right\|^3}$$

Grundlagen
n-Körper-Problem

$$\overrightarrow{a_i(t)} = G \sum_{i \neq j} m_i \frac{r_{ji}(t)}{\left\| r_{ji}(t) \right\|^3}$$

$$Y_o = \begin{pmatrix} m \\ r \\ v \end{pmatrix} \qquad f(Y) = \begin{pmatrix} \dot{m} \\ \dot{r} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 \\ v \\ a \end{pmatrix}$$

Grundlagen
n-Körper-Problem

$$\overrightarrow{a_i(t)} = G \sum_{i \neq j} m_i \frac{r_{ji}(t)}{\left\| r_{ji}(t) \right\|^3}$$

$$Y_o = \begin{pmatrix} m \\ r \\ v \end{pmatrix} \qquad f(Y) = \begin{pmatrix} \dot{m} \\ \dot{r} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 \\ v \\ a \end{pmatrix}$$

Grundlagen
n-Körper-Problem

$$\overrightarrow{a_i(t)} = G \sum_{i \neq j} m_i \frac{r_{ji}(t)}{\left\| r_{ji}(t) \right\|^3}$$

$$Y_o = \begin{pmatrix} m \\ r \\ v \end{pmatrix} \qquad f(Y) = \begin{pmatrix} \dot{m} \\ \dot{r} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 \\ v \\ a \end{pmatrix} \qquad Y_{t+\Delta t} = \begin{pmatrix} m \\ r \\ v \end{pmatrix} = \int_t^{t+\Delta t} f(Y) dY$$

Grundlagen
n-Körper-Problem

$$\overrightarrow{a_i(t)} = G \sum_{i \neq j} m_i \frac{r_{ji}(t)}{\left\| r_{ji}(t) \right\|^3}$$

$$Y_o = \begin{pmatrix} m \\ r \\ v \end{pmatrix} \qquad f(Y) = \begin{pmatrix} \dot{m} \\ \dot{r} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 \\ v \\ a \end{pmatrix}$$

$$Y_{t+\Delta t} = \begin{pmatrix} m \\ r \\ v \end{pmatrix} = \int_t^{t+\Delta t} f(Y)\,dY$$

Durch numerisches Integrieren

Chaos

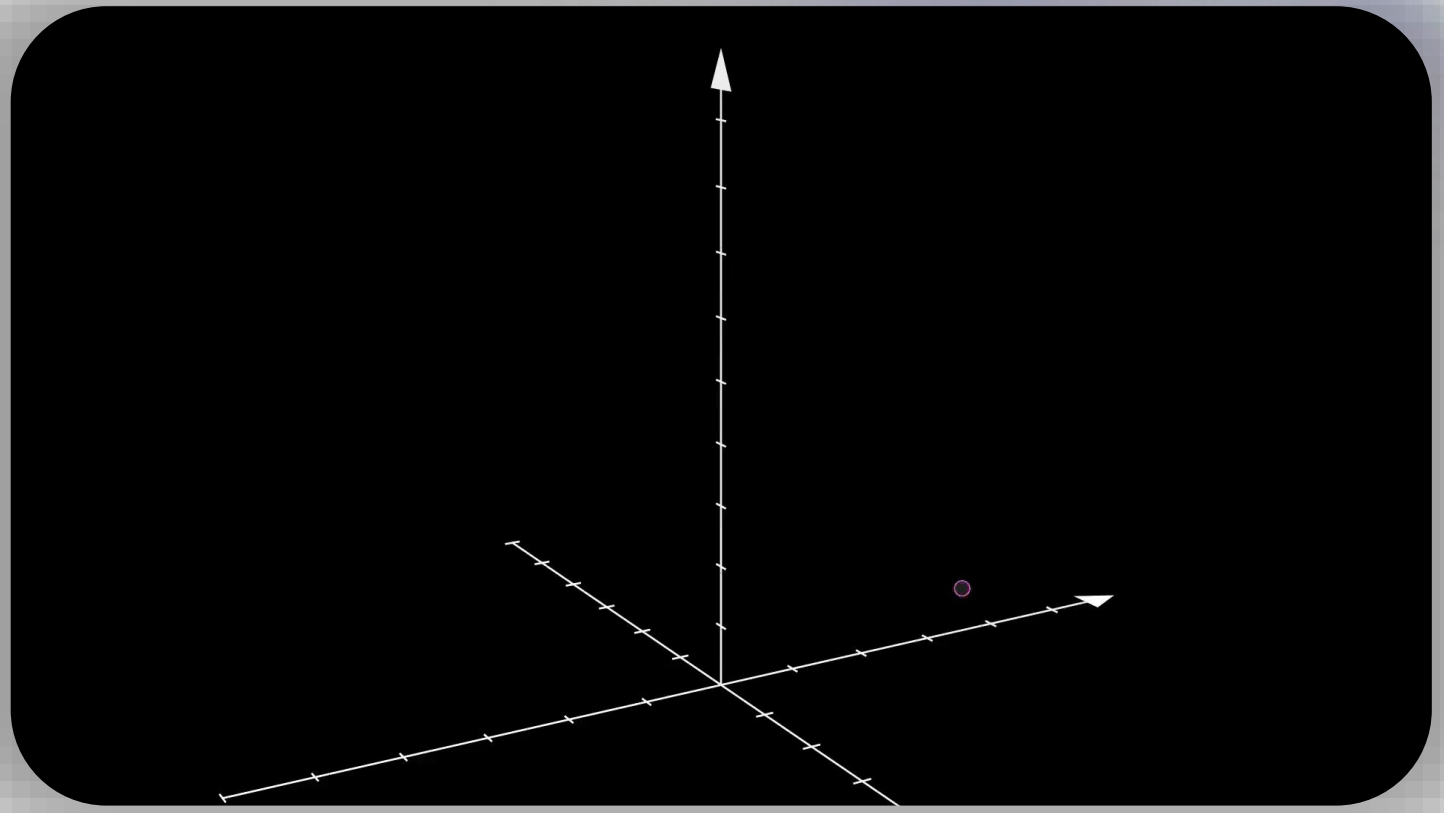
Chaos

Sensitive Abhängigkeit von den Anfangsbedingungen

„Schmetterlingseffekt“

- Kleine Fehler in den Anfangsbedingungen wachsen exponentiell
- Beschreibt Langzeitverhalten in dynamisches System
- Im Allgemeinen nicht Periodisch
- Trotzdem noch Deterministisch
  - Gedankenexperiment: Laplacescher Dämon

Chaos



Beispiel chaotisches System:
Lorenz Attraktor

Chaos
In unserem Sonnensystem

- Ebenfalls n-Körper-System
  - Sonne, Planeten, Asteroiden und Kometen, Einflüsse außerhalb des Sonnensystems
  - Besonderheit: Ein Körper wesentlich größer als alle anderen » Sonne
- Lyapunov-Zeit von ≈ 5 Myr
  - Nach wenigen Lyapunov-Zeiten verschwinden genaue Vorhersagbarkeiten
- Vorhersagen mithilfe von Simulationen

Simulation

```python
G = 6.6743e-11  # Gravitationskonstante in m^3 kg^-1 s^-2
AU_to_m = 1.496e11  # Astronomische Einheit in Meter
AUday_to_ms = AU_to_m / 86400  # AU/Tag in m/s

with open("solar_system.json", "r") as f:
    data = json.load(f)
labels = [
    "Sun",
    "Mercury",
    "Venus",
    "Earth",
    "Mars",
    "Jupiter",
    "Saturn",
    "Uranus",
    "Neptune",
    "Pluto",
    "36 Atalante",
]
rows = []
for name in labels:
    if name in data:
        body = data[name]
        mass = body["mass"]
        pos = body["position"]
        vel = body["velocity"]
        row = [mass, pos["x"], pos["y"], pos["z"], vel["vx"], vel["vy"], vel["vz"]]
        rows.append(row)
# [mass, x, y, z, vx, vy, vz]
bodies = np.array(rows, dtype=np.float64)
bodies[:, 1:4] *= AU_to_m
bodies[:, 4:7] *= AUday_to_ms
```

```python
sim_steps = 0  # Anzahl der bisher berechneten Schritte
sim_time = {0: 0}  # Zeit in Sekunden
paused = False  # Pause-Status
display_index = 0  # Angezeigter History-Eintrag
slider_active = False  # Slider-Aktivität
max_trail_length = 1e4  # Maximale Länge der Trails

# History zur Speicherung der Zustände (für Slider und Trails)
history = [[bodies[i, 1:4].copy()] for i in range(bodies.shape[0])]

# Plot Setup (angepasster unterer Rand für Slider/Buttons)
fig = plt.figure(figsize=(25, 25), dpi=100)
ax = fig.add_subplot(111, projection="3d")
plt.subplots_adjust(left=0.1, right=0.95, top=0.95, bottom=0.1)

ax.set_xlim(-axis_limit, axis_limit)
ax.set_ylim(-axis_limit, axis_limit)
ax.set_zlim(-axis_limit, axis_limit)
ax.set_box_aspect((1, 1, 1))

# Planeten (Punkte) und Trails (Linien) zeichnen
planets = [ax.plot([], [], [], "o", color=f"C{i}", markersize=10, alpha=0.8, label=labels[i])[0] for i in range(bodies.shape[0])]
trails = [ax.plot([], [], [], "-", color=f"C{i}", alpha=0.3, linewidth=1)[0] for i in range(bodies.shape[0])]

# Zeitanzeige (als Text im Plot)
time_text = ax.text2D(0.05, 0.95, "", transform=ax.transAxes)

# Legende
ax.legend(loc="upper right")
```

```python
def update(frame):
    global bodies, sim_steps, display_index
    bodies_new, corrected_dt = rkdp45(bodies, dt)
    bodies[:] = bodies_new
    sim_steps += 1
    sim_time[sim_steps] = sim_time.get(sim_steps - 1, 0) + corrected_dt

    for i in range(bodies.shape[0]):
        history[i].append(bodies[i, 1:4].copy())

    # Update Slider-Bereich dynamisch
    slider.valmax = sim_steps
    slider.ax.set_xlim(slider.valmin, slider.valmax)

    # Automatische Slider-Bewegung nur wenn nicht pausiert
    if not paused and not slider_active:
        display_index = sim_steps
        slider.set_val(display_index)

    update_display(display_index)
    return planets + trails + [time_text]


ani = FuncAnimation(fig, update, interval=0, cache_frame_data=False)
plt.show()
```

```python
@nb.njit(fastmath=True, parallel=True)
def rkdp45(bodies, dt):
    # c-Werte (Zeitanteile)
    c2, c3, c4, c5, c6, c7 = 1 / 5, 3 / 10, 4 / 5, 8 / 9, 1.0, 1.0

    # Butcher-Tabellen-Koeffizienten:
    a21 = 1 / 5
    a31, a32 = 3 / 40, 9 / 40
    a41, a42, a43 = 44 / 45, -56 / 15, 32 / 9
    a51, a52, a53, a54 = 19372 / 6561, -25360 / 2187, 64448 / 6561, -212 / 729
    a61, a62, a63, a64, a65 = 9017 / 3168, -355 / 33, 46732 / 5247, 49 / 176, -5103 / 18656
    a71, a72, a73, a74, a75, a76 = 35 / 384, 0.0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84

    # Koeffizienten für 5. Ordnung (5th-Order Lösung):
    b1, b2, b3, b4, b5, b6 = 35 / 384, 0.0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84
    # b7 = 0.0 implizit

    # Koeffizienten für die 4. Ordnung (eingebettete Lösung):
    b1s, b2s, b3s, b4s, b5s, b6s, b7s = 5179 / 57600, 0.0, 7571 / 16695, 393 / 640, -92097 / 339200, 187 / 2100, 1 / 40

    k1 = acceleration(bodies)
    k2 = acceleration(bodies + dt * (a21 * k1))
    k3 = acceleration(bodies + dt * (a31 * k1 + a32 * k2))
    k4 = acceleration(bodies + dt * (a41 * k1 + a42 * k2 + a43 * k3))
    k5 = acceleration(bodies + dt * (a51 * k1 + a52 * k2 + a53 * k3 + a54 * k4))
    k6 = acceleration(bodies + dt * (a61 * k1 + a62 * k2 + a63 * k3 + a64 * k4 + a65 * k5))
    k7 = acceleration(bodies + dt * (a71 * k1 + a72 * k2 + a73 * k3 + a74 * k4 + a75 * k5 + a76 * k6))

    y5 = bodies + dt * (b1 * k1 + b2 * k2 + b3 * k3 + b4 * k4 + b5 * k5 + b6 * k6)
    y4 = bodies + dt * (b1s * k1 + b2s * k2 + b3s * k3 + b4s * k4 + b5s * k5 + b6s * k6 + b7s * k7)
    error = np.linalg.norm(y5 - y4)

    print("dt:", dt, "error:", error)
    if error > tolerance:
        dt = 0.99 * dt * (tolerance / error) ** 0.01
        return rkdp45(bodies, dt)
    return y5, dt
```

```python
@nb.njit(fastmath=True, parallel=True)
def euler(bodies, dt):
    acc = acceleration(bodies)
    bodies[:, 1:4] += dt * bodies[:, 4:7]
    bodies[:, 4:7] += dt * acc[:, 4:7]
    return bodies, dt
```

```python
@nb.njit(fastmath=True, parallel=True)
def rk4(bodies, dt):
    k1 = acceleration(bodies)
    k2 = acceleration(bodies + dt * 0.5 * k1)
    k3 = acceleration(bodies + dt * 0.5 * k2)
    k4 = acceleration(bodies + dt * k3)

    bodies[:, 1:4] += dt * (bodies[:, 4:7] + dt * (k1[:, 4:7] + 2 * k2[:, 4:7] + 2 * k3[:, 4:7] + k4[:, 4:7]) / 6)
    bodies[:, 4:7] += dt * (k1[:, 4:7] + 2 * k2[:, 4:7] + 2 * k3[:, 4:7] + k4[:, 4:7]) / 6

    return bodies, dt
```

Simulation

Hier Simulation vom Sonnensystem und ggf. abstrakten 3-Körper-System zeigen

Simulation
Symplektische Integratoren

- Basieren auf Hamiltonschen Systemen
- Erhalten möglichst gut Energie und Phasenraumvolumen über lange Zeit
- Beispiele: Yoshida, Verlet, WHFast

```python
@nb.njit(fastmath=True, parallel=True)
def verlet(bodies, dt):
    for i in range(bodies.shape[0]):
        # Calculate acceleration for current position
        ai = np.zeros(3)
        for j in range(bodies.shape[0]):
            if i == j:
                continue
            r = bodies[j, 1:4] - bodies[i, 1:4]
            dist = np.sqrt(np.sum(r * r) + 1e-12)
            ai += G * bodies[j, 0] * r / (dist * dist * dist)

        # First kick
        bodies[i, 4:7] += 0.5 * dt * ai

        # Drift
        bodies[i, 1:4] += dt * bodies[i, 4:7]

        # Recalculate acceleration for new position
        ai2 = np.zeros(3)
        for j in range(bodies.shape[0]):
            if i == j:
                continue
            r = bodies[j, 1:4] - bodies[i, 1:4]
            dist = np.sqrt(np.sum(r * r) + 1e-12)
            ai2 += G * bodies[j, 0] * r / (dist * dist * dist)

        # Second kick
        bodies[i, 4:7] += 0.5 * dt * ai2

    return bodies, dt
```

Vorhersagen

- Laskar 2008
  - Statistische Wahrscheinlichkeiten für Instabilitäten
  - Einfluss von relativistischen Effekten
  - 1001 Durchführen
- Laskar 2012
  - Erklärt historischen Kontext
  - Zusammenfassung früherer Studien und deren Aussagen

Vorhersagen

- Brown & Rein 2020
  - Öffentlicher Datensatz zu 96 Durchführungen
  - Volle N-Körper-Integration mit Open-Source-Tools
- Brown & Rein 2022
  - 2880 Durchführungen mit Open-Source Code
  - Untersuchung von „stellare fly-bys“
    - Auswirkung von veränderter Bahn Neptuns auf innere Planeten

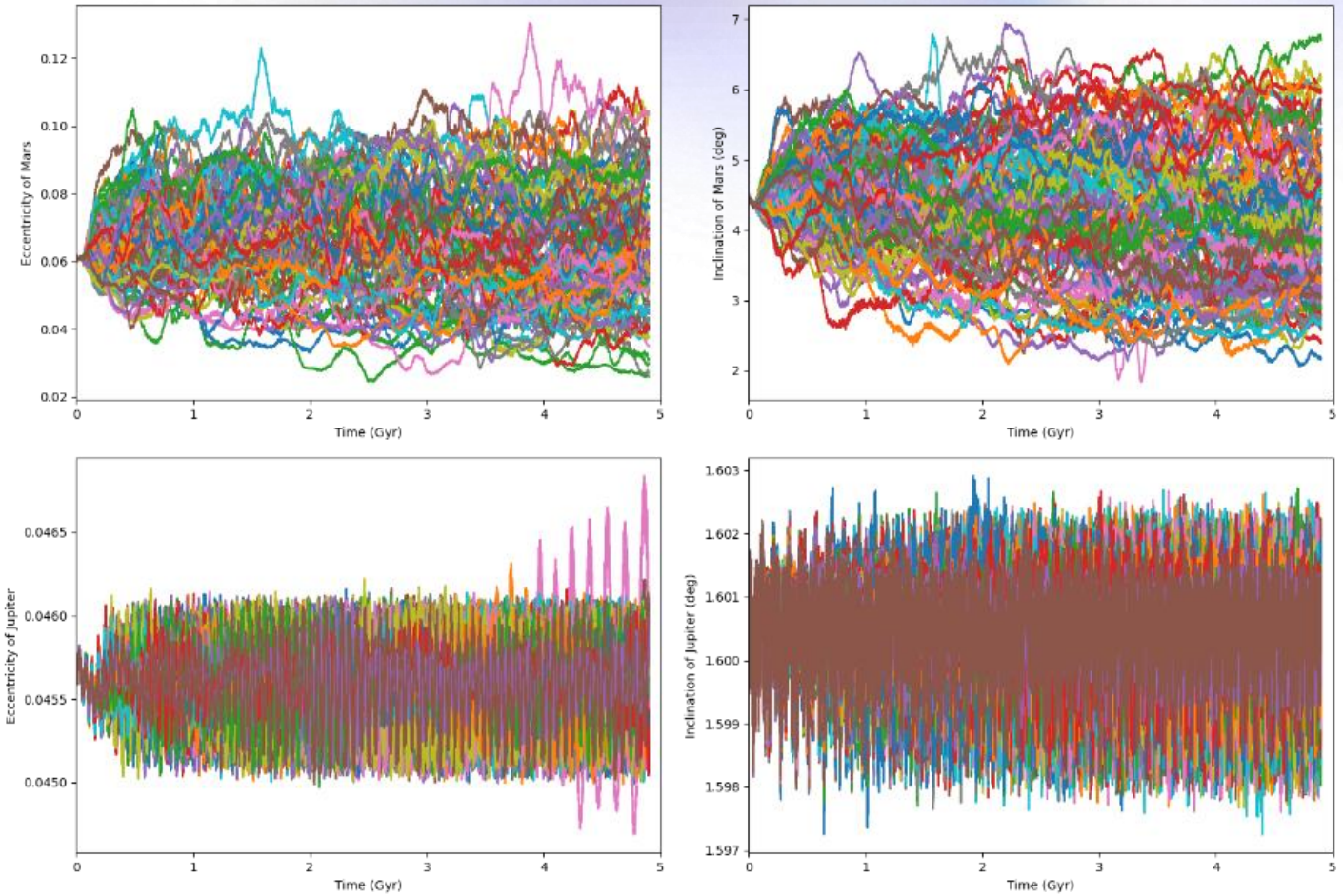
Vorhersagen

Innere & Äußere Planeten

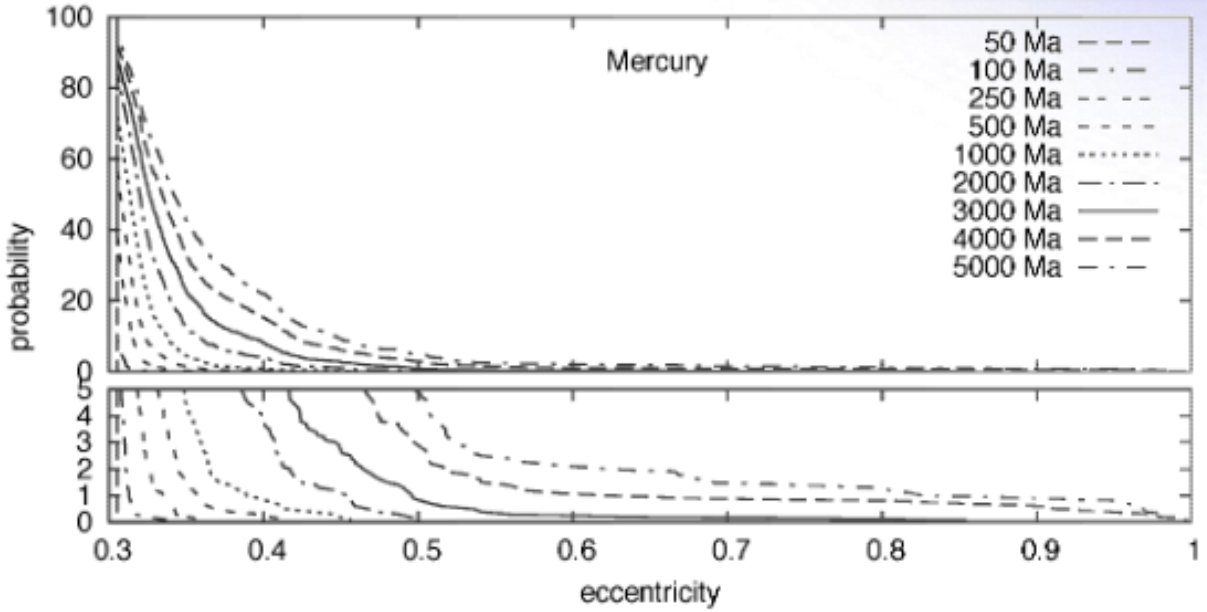
Gyr interval. As it was mentioned before, (Laskar, 1990, 1994), there is practically no diffusion for the outer planet system that behaves nearly as a quasiperiodic and regular system. On the opposite, there is a significant diffusion of the eccentricities and inclinations of the inner planets. The statistics on the maximum values reached by the ec-

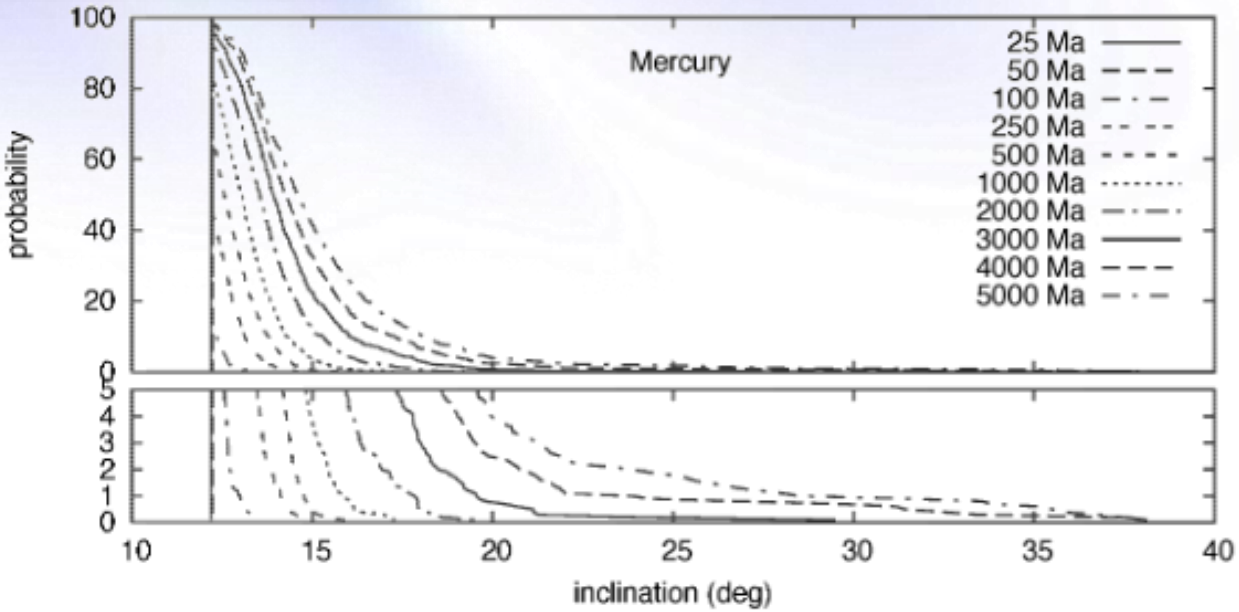
Vorhersagen

Innere & Äußere Planeten



Vorhersagen
Merkur

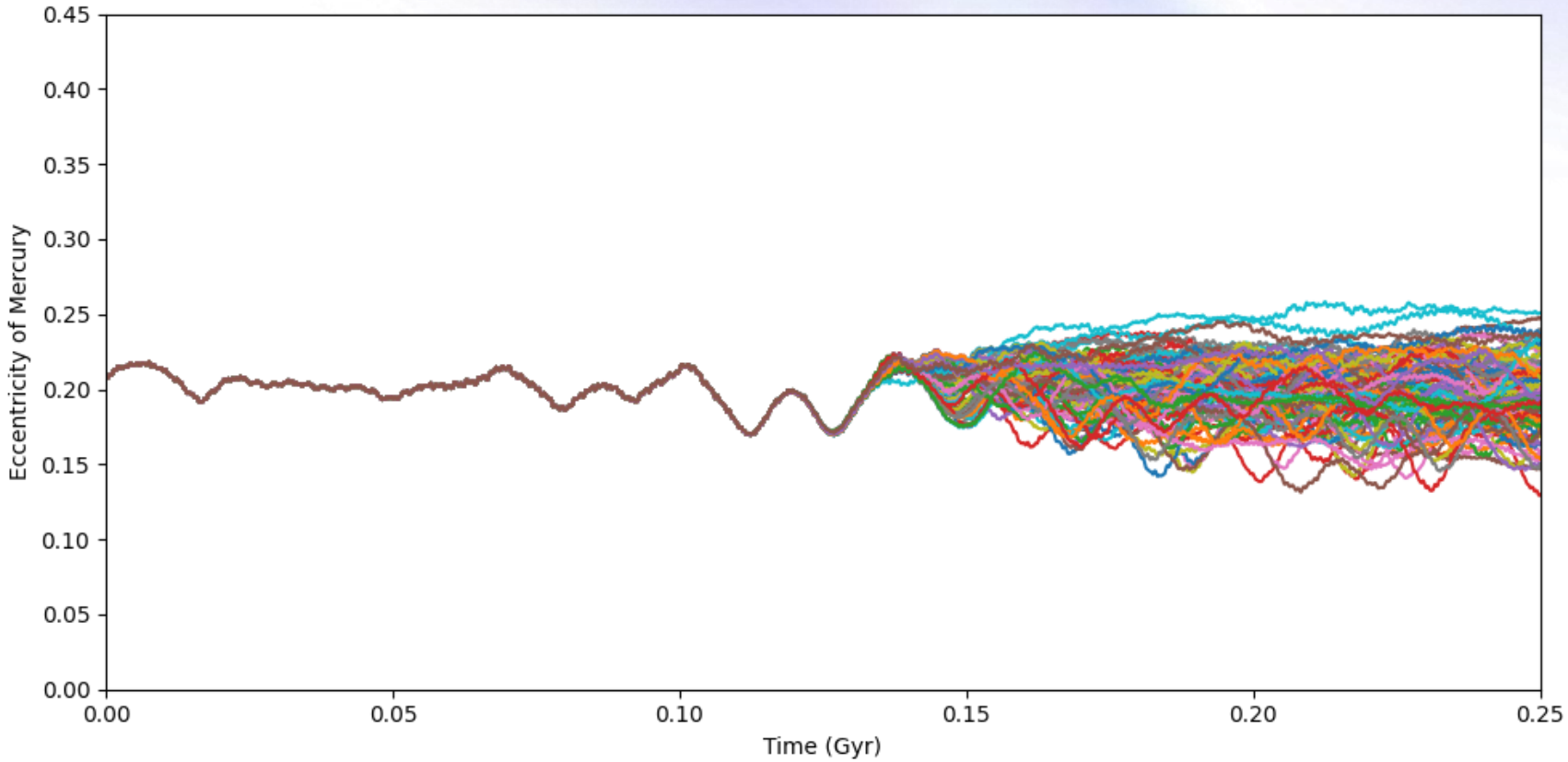




$e \gtrsim 0.8 \quad \rightarrow Gefahr\ Kollision\ mit\ Venus$
$e \gtrsim 0.95 \rightarrow Gefahr\ Kollision\ mit\ Sonne$
$e > 1 \quad \rightarrow Rauswurf$

Abb. 3: Laskar Diagramme der inneren Planeten

Vorhersagen
Merkur



Vorhersagen
Newton vs. Relativität

Newtonsche Mechanik

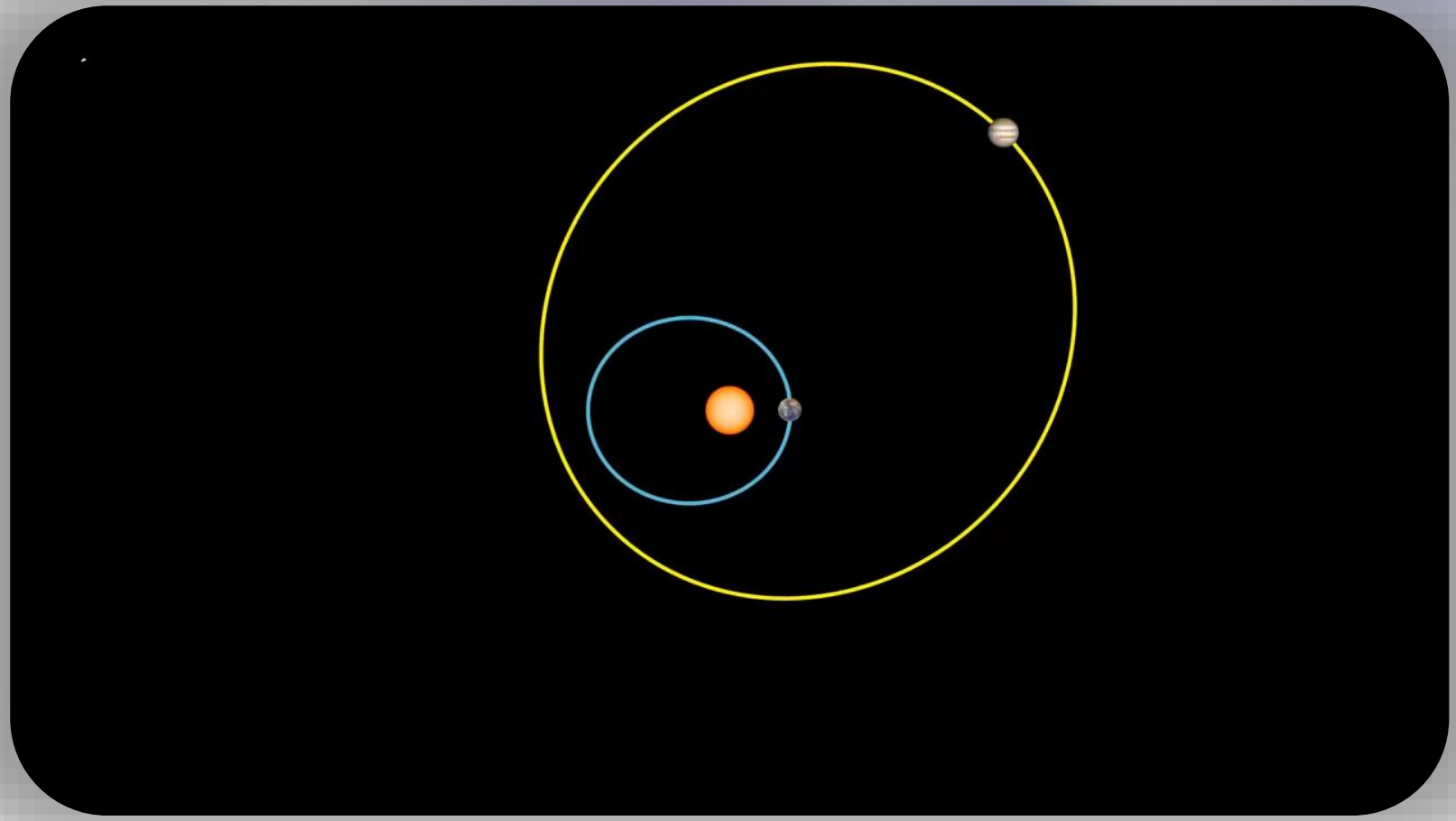
e_{m0}	500	1000	1500	2000	3000	4000	5000
0.35	130	341	478	558	692	763	812
0.40	75	249	373	449	589	684	747
0.50	24	118	226	306	442	552	640
0.60	16	76	169	238	364	476	564
0.70	14	67	150	218	343	454	541
0.80	12	63	141	209	331	442	531
0.90	12	61	138	202	325	441	530

Allgemeine Relativitätstheorie

e_{m0}	500	1000	1500	2000	3000	4000	5000
0.35	25	75	128	165	280	366	427
0.40	4	21	38	52	113	180	243
0.50	0	0	0	0	6	19	33
0.60	0	0	0	0	0	6	10
0.70	0	0	0	0	0	6	10
0.80	0	0	0	0	0	2	8
0.90	0	0	0	0	0	0	2

Abb. 4: Laskar Diagramme – Vergleich Newton vs. GR

Vorhersagen
Säkulare Resonanzen



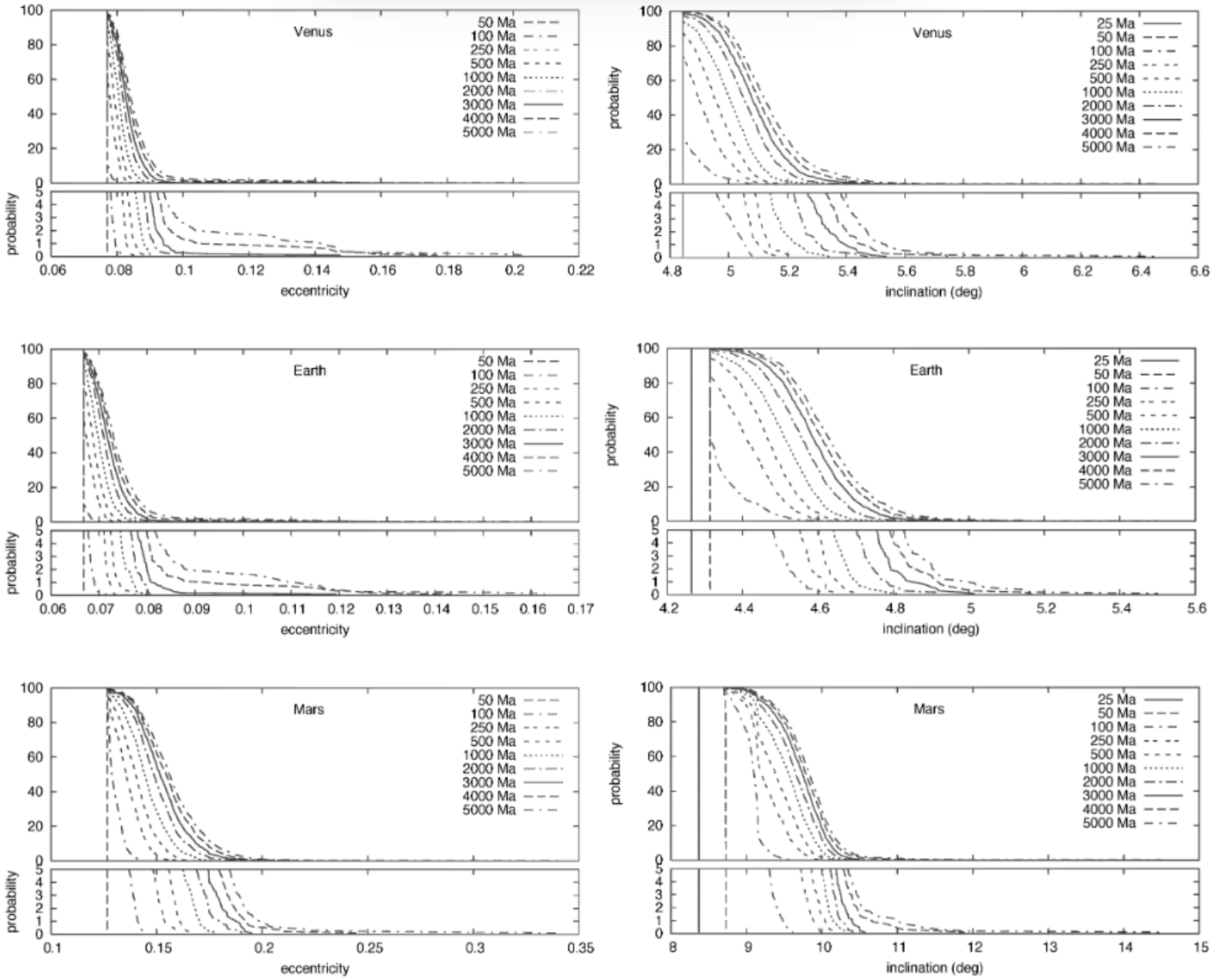


Abb. 3:
Laskar Diagramme
der inneren Planeten

Vorhersagen
Auf einen Blick

- Chaotische Effekte setzen nach ≈ 50-100 Myr ein
- Kollision Merkurs ≈ 1% nach 5 Gyr
  - Ohne relativistische Effekte ≈ 60%
  - Hauptsächlich durch Merkur-Jupiter-Resonanz
- Kollision Mars-Erde sehr unwahrscheinlich
  - < 0,2%
- Einfluss von anderen Sternen nicht zu vernachlässigen
  - Geringe Chance eines „stellar fly-bys“ in 5 Gyr

Weitere Zukunftsaussichten

Tod der Sonne – Sonne als Roter Riese

- Wasserstoffbrennende Phase endet in ≈ 5 (±0,5) Gyr
  - Vollständige Verbrauch von Wasserstoffatomen
  - Maximum von vielen Simulationen
- Anstieg des Radius auf ≈ 0,75 AE
  - Entspricht der Umlaufbahn von Venus
- Verliert über Sonnenwinde ca. 28% ihrer Masse

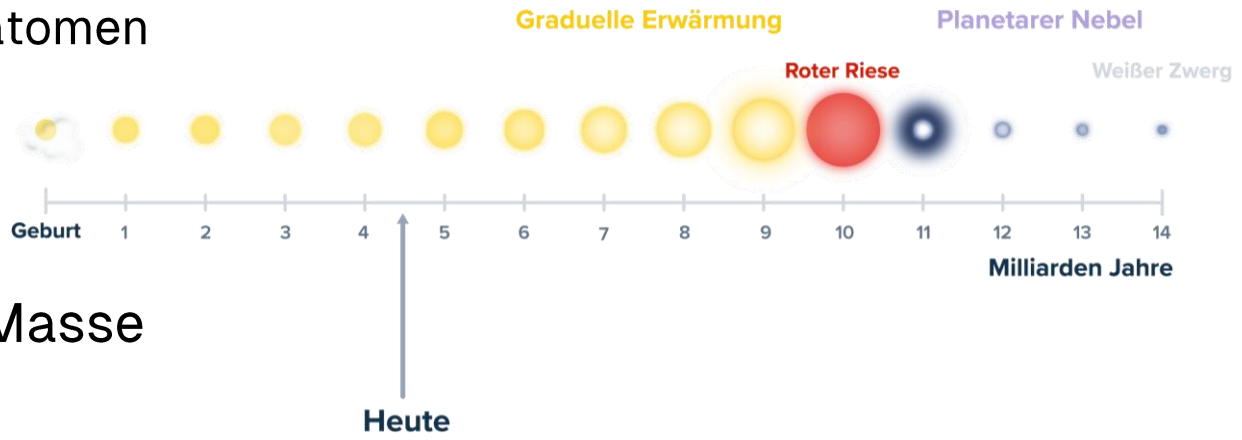


Abb. 5: Lebenszyklus der Sonne

Weitere Zukunftsaussichten
Asteroiden und Kometen

- Lyapunov-Zeiten von 100 – 100.000 Jahren
- Für die Erde gefährliche Objekte früh genug erkannt
- Kollisionen mit Planeten können Simulationen verfälschen

- Einschlaghäufigkeiten
  - "$d_{Objekt} \geq 1km$" ca. alle 0,5 Myr
  - "$d_{Objekt} \approx 5km$" ca. alle 20 Myr
  - "$d_{Objekt} \approx 10km$" ca. alle 100 Myr

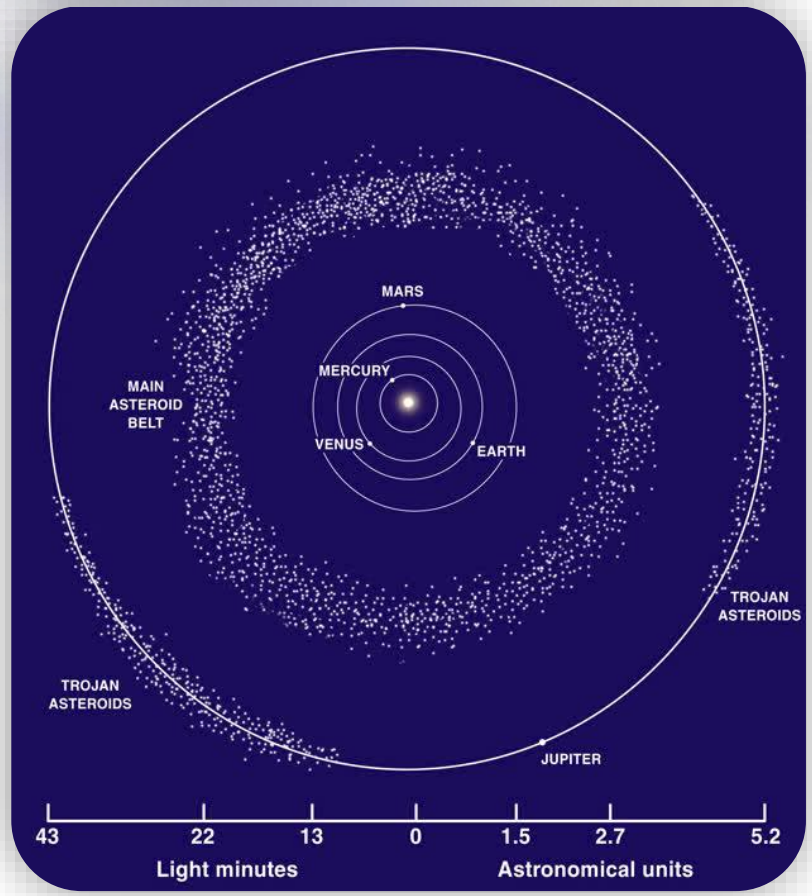


Abb. 2: Asteroidengürtel & Positionen

Fazit

Wo liegen die Grenzen der Vorhersagbarkeit?

Fazit

Nächste ≈ 100 Millionen Jahre

- Quasiperiodische Planetenbahnen
  - Keine Kollisionen & Auswürfe
- Kleine Unsicherheiten von Asteroiden & Kometen
  - Allerdings > 96% aller NEAs bereits bekannt
  - Ablenkbar und Berechenbar
- Hohe Vorhersagbarkeit in naher Zukunft

100 Millionen – 5 Milliarden Jahre

- Chaotische Effekte
  - Nur noch statistische Aussagen möglich
  - 0,38mm Änderung
    -> völlig andere Bahnen nach ≈ 200 Myr
- Unvorhersagbare Sterneneinflüsse
- Lebenszyklus der Sonne sicher bekannt
- Vorhersagbarkeit nimmt mit der Zeit stark ab

Quellen

- https://doi.org/10.48550/arXiv.2012.05177
- https://doi.org/10.48550/arXiv.0802.3371
- https://doi.org/10.48550/arXiv.1209.5996
- https://doi.org/10.48550/arXiv.1705.00527
- https://doi.org/10.48550/arXiv.1506.01084
- https://zenodo.org/records/4299102
- https://doi.org/10.1051/0004-6361/202140989

- https://orbital-mechanics.space/
- https://rebound.readthedocs.io/en/latest/integrators/#whfast
- https://eyes.nasa.gov/apps/solar-system/
- https://science.nasa.gov/solar-system/
- https://ssd.jpl.nasa.gov/horizons/app.html

Quellen

- https://en.wikipedia.org/wiki/Ergodicity
- https://en.wikipedia.org/wiki/Two-body_problem
- https://en.wikipedia.org/wiki/Three-body_problem
- https://en.wikipedia.org/wiki/Liouville%E2%80%93Arnold_theorem
- https://en.wikipedia.org/wiki/Orbital_resonance
- https://en.wikipedia.org/wiki/Stability_of_the_Solar_System
- https://en.wikipedia.org/wiki/Lagrange_point
- https://en.wikipedia.org/wiki/Newton%27s_law_of_universal_gravitation
- https://en.wikipedia.org/wiki/Orbital_period
- https://en.wikipedia.org/wiki/Lyapunov_time
- https://en.wikipedia.org/wiki/Dormand%E2%80%93Prince_method
- https://en.wikipedia.org/wiki/Bessel_function
- https://en.wikipedia.org/wiki/Kepler%27s_equation
- https://en.wikipedia.org/wiki/Ellipse
- https://en.wikipedia.org/wiki/Orbit_equation
- https://en.wikipedia.org/wiki/Newton%27s_method
- https://en.wikipedia.org/wiki/Hamiltonian_mechanics
- https://en.wikipedia.org/wiki/Verlet_integration
- https://en.wikipedia.org/wiki/Numerical_integration
- https://en.wikipedia.org/wiki/Periodic_function

Quellen

- Abb. 1: https://commons.wikimedia.org/wiki/File:Kepler%27s_equation_scheme_German.svg
- Abb. 2: https://www.spektrum.de/news/asteroidenguertel-um-sonne-gesteinsbrocken-zwischen-jupiter-und-mars/982787
- Abb. 3 & 4: https://doi.org/10.48550/arXiv.0802.3371
- Abb. 5: https://www.studysmarter.de/schule/physik/astronomie/sonne/