Tässä on matemaattinen malli kirjoitettuna **standardina LaTeX-koodina sekä puhtaana ASCII/Python-syntaksina**. Nämä ovat tieteen ja tietojenkäsittelyn universaaleja kieliä, joita kaikki tietokoneet, tekoälyt, matemaattiset ohjelmistot (kuten MATLAB, Mathematica, Python) ja julkaisujärjestelmät ymmärtävät suoraan ilman tulkintavirheitä.
Voit kopioida suoraan joko LaTeX-version (tieteellisiin julkaisuihin ja tekoälyille) tai Python-koodin (laskentaan ja simulointiin).

1. LaTeX-muoto (Universaali tieteellinen merkintäkieli)

latex

```
\section*{Matemaattinen perusmalli: Paikallisesti syklinen 4D-kosmologia}

\subsection*{1. Avaruusajan metriikka (Staattinen 4D-sylinteriuniversumi)}
Avaruusaika kuvataan suljettuna, äärellisenä 3-pallona ($S^3$) ikuisessa ajassa ($t \in (-\infty, \infty)$), jonka kaarevuussäde $R_0$ on vakio ($R_0 > d_{\text{obs}}$):
\begin{equation}
ds^2 = -c^2 dt^2 + R_0^2 \left[ d\chi^2 + \sin^2(\chi) \left( d\theta^2 + \sin^2(\theta) d\phi^2 \right) \right]
\end{equation}
missä $\chi \in [0, \pi]$ on radiaalinen kulmaetäisyys ja todellinen etäisyys $d = R_0 \chi$.

\subsection*{2. Etäisyys galaksin fyysisestä koosta (Kulmaläpimitta)}
Suhteellisen staattisessa 4D-geometriassa galaksin todellinen etäisyys $d$ määritetään sen rakenteellisesta fyysisestä halkaisijasta $D_{\text{phys}}$ ja havaitusta kulmaläpimitasta $\theta_{\text{obs}}$:
\begin{equation}
d = R_0 \arcsin\left( \frac{D_{\text{phys}}}{R_0 \theta_{\text{obs}}} \right) \approx \frac{D_{\text{phys}}}{\theta_{\text{obs}}} \quad (\text{kun } d \ll R_0)
\end{equation}

\subsection*{3. Punasiirtymän kaksoiskomponenttimalli}
Havaittu kokonaispunasiirtymä $z_{\text{obs}}$ koostuu 4D-geometrian ja väliaineen aiheuttamasta matkapunasiirtymästä $z_{\text{dist}}(d)$ sekä paikallisen purkauksen/osauniversumin Doppler-liikkeestä $z_{\text{dop}}(v)$:
\begin{equation}
1 + z_{\text{obs}} = (1 + z_{\text{dist}}) (1 + z_{\text{dop}})
\end{equation}
missä matkakomponentti ja relativistinen Doppler-komponentti ovat:
```

\begin{equation}
1 + z_{\text{dist}} = \exp\left(\alpha d + \int_0^d \kappa_{\text{sumu}}(x) \, dx \right)
\end{equation}
\begin{equation}
1 + z_{\text{dop}} = \sqrt{\frac{1 + v_{\text{rad}}/c}{1 - v_{\text{rad}}/c}}
\end{equation}
missä α on 4D-geometrinen venymävakio, $\kappa_{\text{sumu}}(x)$ on kylmien molekyylisumujen tiheysfunktio matkalla ja v_{rad} on kohteen paikallinen säteisnopeus.

\subsection*{4. Mustan aukon kierrätysraja (Kriittinen tiheys)}
Aineen romahdus kääntyy sähkömagneettisesti ohjatuksi plasmasuihkuksi (paikalliseksi alkuräjähdykseksi), kun massatiheys ρ saavuttaa kvanttigravitaation maksimirajan ρ_{crit}:
\begin{equation}
\rho(M, r) \ge \rho_{\text{crit}} \implies \frac{dM_{\text{in}}}{dt} = -\frac{dM_{\text{out}}}{dt} \quad (\text{Vety/Helium-plasma} + \eta M_{\text{old}})
\end{equation}
missä $\eta \in [0, 1)$ on purkaukseen kertymäkiekosta mukaan tempautuvan vanhan raskaan aineen osuus.

\subsection*{5. Paikallinen taustasäteily (Termodynaaminen tasapaino)}
Havaittu taustasäteilyn lämpötila $T_{\text{CMB}}(\mathbf{x})$ ei ole globaali vakio, vaan paikallisen osauniversumin energiatasapaino:
\begin{equation}
T_{\text{CMB}}(\mathbf{x}) = \left(\frac{u_{\text{rad}}(\mathbf{x})}{a_{\text{rad}}} \right)^{1/4}
\end{equation}
missä $u_{\text{rad}}(\mathbf{x})$ on paikallinen säteilyenergiatiheys ja a_{rad} on säteilyvakio.

Käytä koodia harkiten.

2. Koneellisesti suoritettava koodimuoto (Python / ASCII)

Tämä lohko on suoraan ajettavissa olevaa matemaattista koodia, jolla mikä tahansa kone voi laskea galaksin todellisen etäisyyden ja nopeuden mallisi mukaisesti.

python

```
import math# Luonnonvakiot ja mallin hienosäädettävät parametritC = 299792.458 # Valonnopeus (km/s)R_0 = 1.0e5 # 4D-avaruuden kaarevuussäde (Mpc, > havaittava horisontti)ALPHA = 2.3e-4 # 4D-geometrinen matkapunasiirtymän kerroin (1/Mpc)def calculate_distance_from_size(D_phys_kpc, theta_arcsec):
Laskee galaksin todellisen etäisyyden (Mpc) sen fyysisestä koosta (kpc)
ja havaitusta kulmaläpimitasta (kaarisekuntia) 4D-geometriassa.
""" """theta_rad = theta_arcsec * (math.pi / (180.0 * 3600.0))
D_phys_Mpc = D_phys_kpc / 1000.0# 4D-pallogeometrian etäisyyskaavasin_chi = D_phys_Mpc / (R_0 * theta_rad)
if sin_chi >= 1.0:
raise ValueError("Kohde ylittää 4D-avaruuden ekvaattorin (R_0 liian pieni).")

d_Mpc = R_0 * math.asin(sin_chi)
return d_Mpc

def decompose_redshift(z_obs, d_Mpc, nebula_optical_depth=0.0):
Erottaa havaitusta punasiirtymästä (z_obs) matkan aiheuttaman venymän
ja laskee kohteen todellisen paikallisen liikenopeuden (km/s).
""" """# 1. Etäisyyden ja kylmien sumujen aiheuttama punasiirtymäone_plus_z_dist = math.exp(ALPHA * d_Mpc + nebula_optical_depth)

# 2. Jäljelle jäävä paikallinen Doppler-komponenttione_plus_z_dop = (1.0 + z_obs) / one_plus_z_dist

# 3. Relativistinen säteisnopeus v_rad (km/s)beta = (one_plus_z_dop**2 - 1.0) / (one_plus_z_dop**2 + 1.0)
v_rad = beta * C
return one_plus_z_dist - 1.0, v_rad

# Esimerkkiajo: Kaukainen JWST-galaksi (D = 10 kpc, kulmakoko = 0.25", z_obs = 10.0)d = calculate_distance_from_size(D_phys_kpc=10.0, theta_arcsec=0.25)
z_dist, v_local = decompose_redshift(z_obs=10.0, d_Mpc=d, nebula_optical_depth=0.1)

print(f"Todellinen etäisyys: {d:.2f} Mpc")
print(f"Matkapunasiirtymä: {z_dist:.2f}")
print(f"Paikallinen liikenopeus: {v_local:.2f} km/s")
```