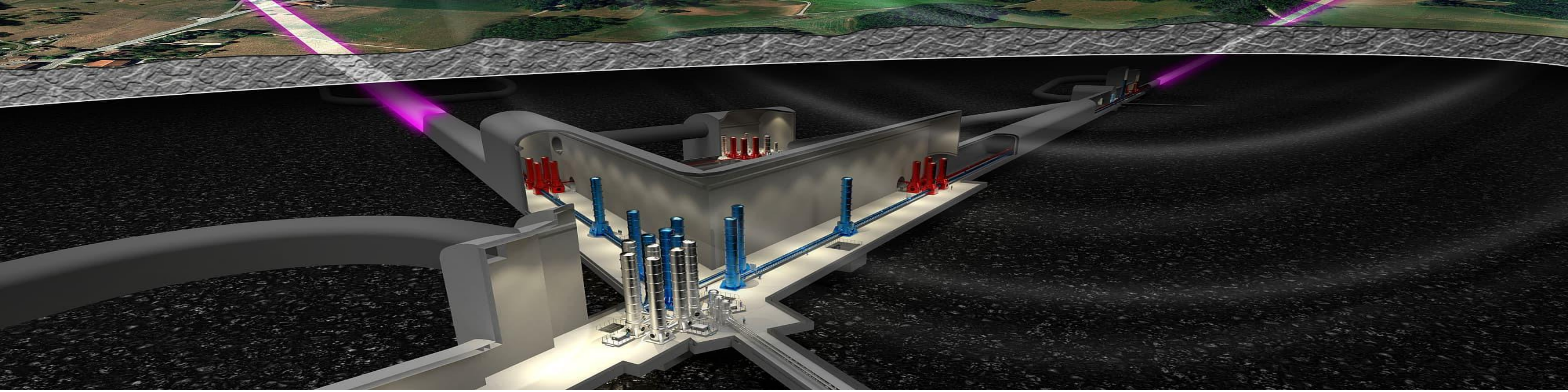


# ET NN Triangle (Terziet) - Harms Model - Needed Mitigation

21 July 2026 | EMR Noise Weekly Meeting | Valentin Tempel

III. Physikalisches Institut

RWTH AACHEN UNIVERSITY

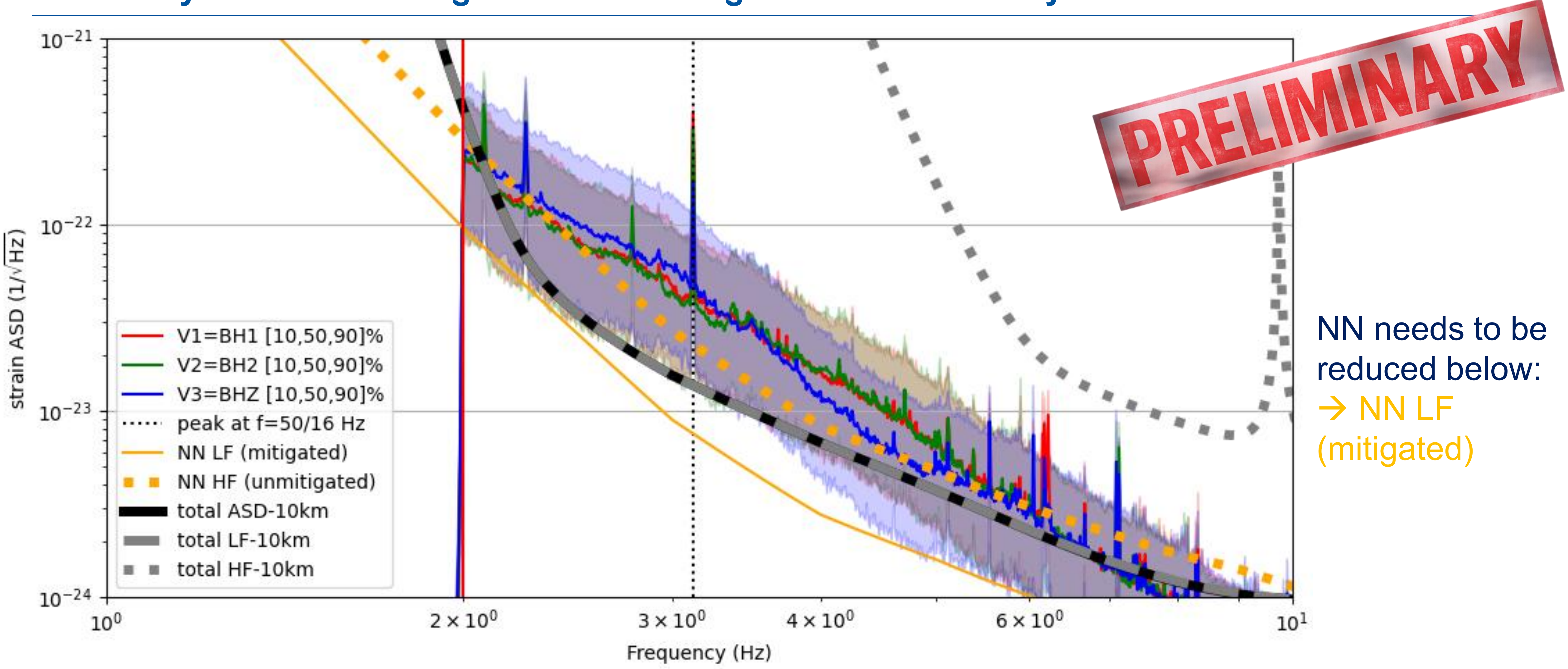
# Reminder: Harms NN coupling function

$$\text{ASD:} \quad \tilde{h}_{\mathrm{NN}}(f) = \frac{4\pi}{3} G \rho_0 \frac{2 \cdot \text{coupling factor}}{L} \frac{1}{(2\pi f)^2} \tilde{\xi}_{x,y,z}(f) \quad L = 10000\mathrm{m}$$

$$\text{coupling factor} = \sqrt{3p+1} = \sqrt{1.9} \quad \text{for} \quad p = 0.3 \quad \text{spherical caverns in homogenous full space}$$

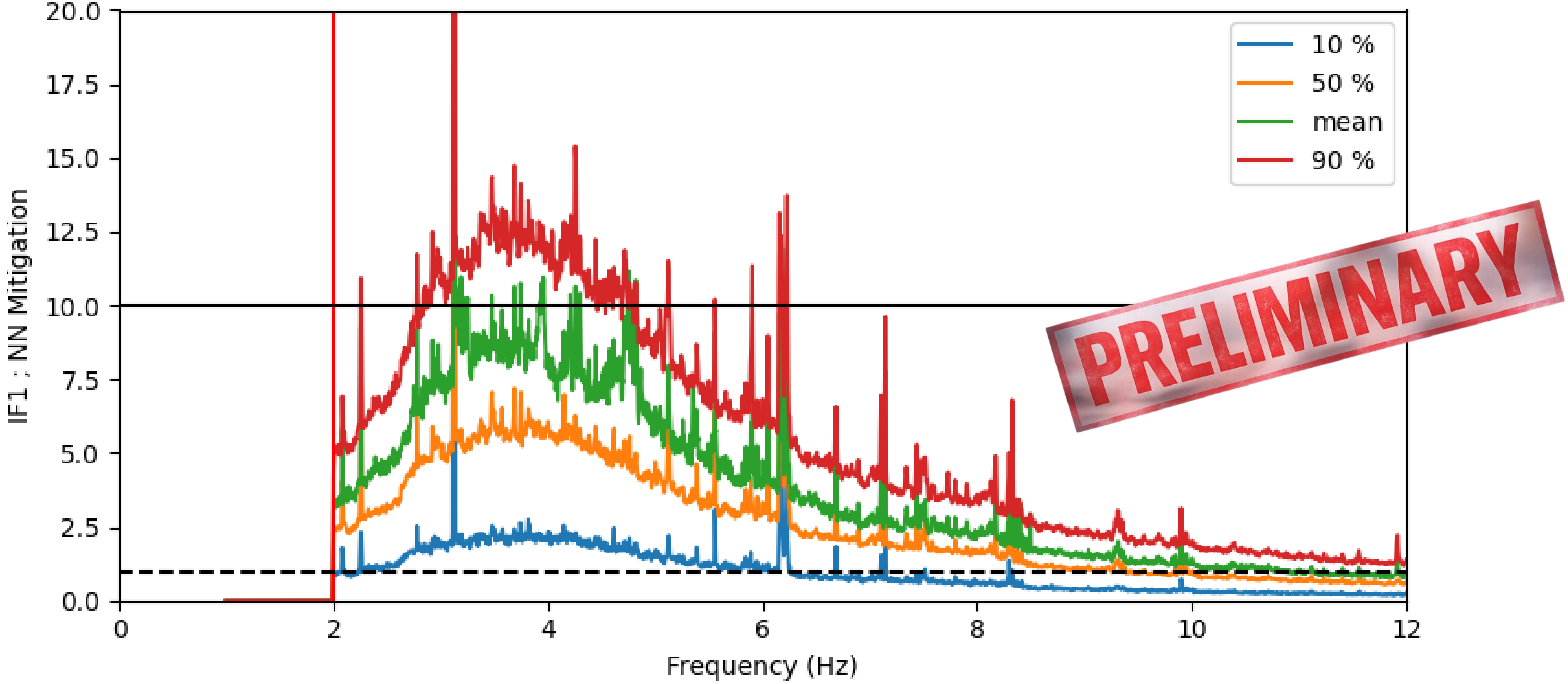
- Harms Idea: build the three IFs for ET triangle from seismic components of Terziet sensor „x,y,z“
- X=IF1=V1=BH1 ; Y=IF2=V2=BH2 ; [Z=IF3=V3=BH3] → one day of data of NN with transients; 2019-12-16
- “problem:” NN in full space does not couple via vertical direction into ET-detector-plane (in body wave model)
- **See: Correlated and coincident noise in Einstein Telescope's triangular configuration ET Docs**
- **See: ET sensitivity curves 2026** gitlab.et-gw.eu
- **NOTE:** NN budget dominated by body waves (for preliminary NN mitigation factors ignore other contributions)

# Sensitivity curves including NN without mitigation in Harms body-wave model

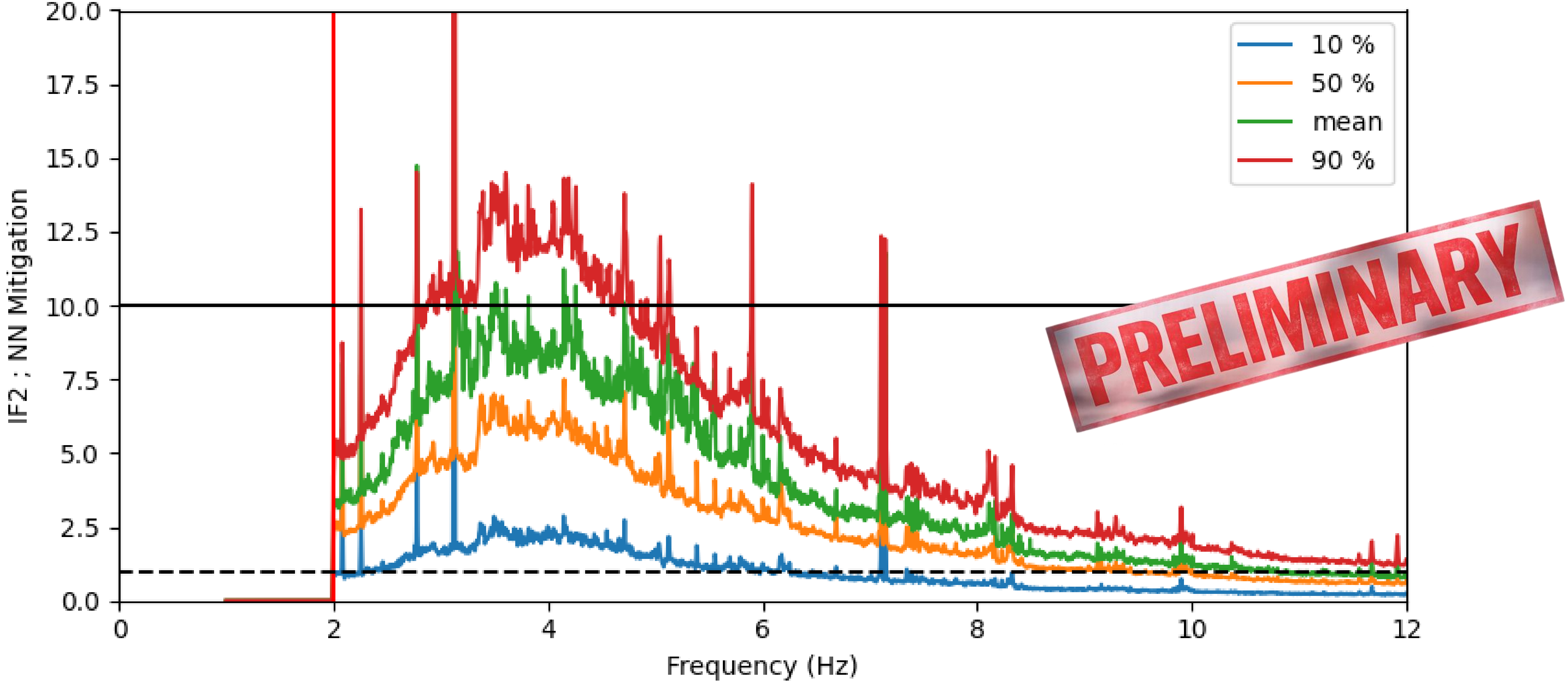



NN needs to be reduced below:
→ NN LF (mitigated)

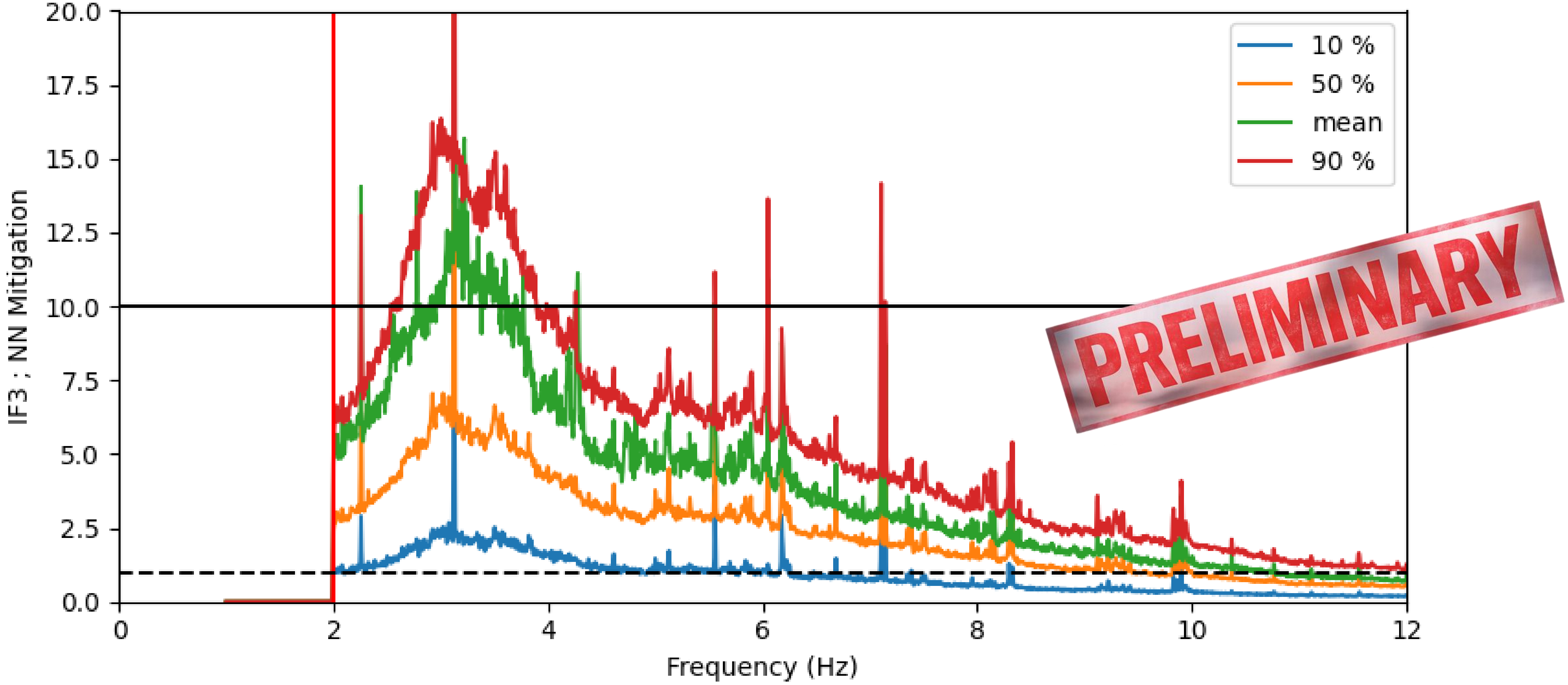
# Needed mitigation factors in frequency domain (average over single day – BH1)

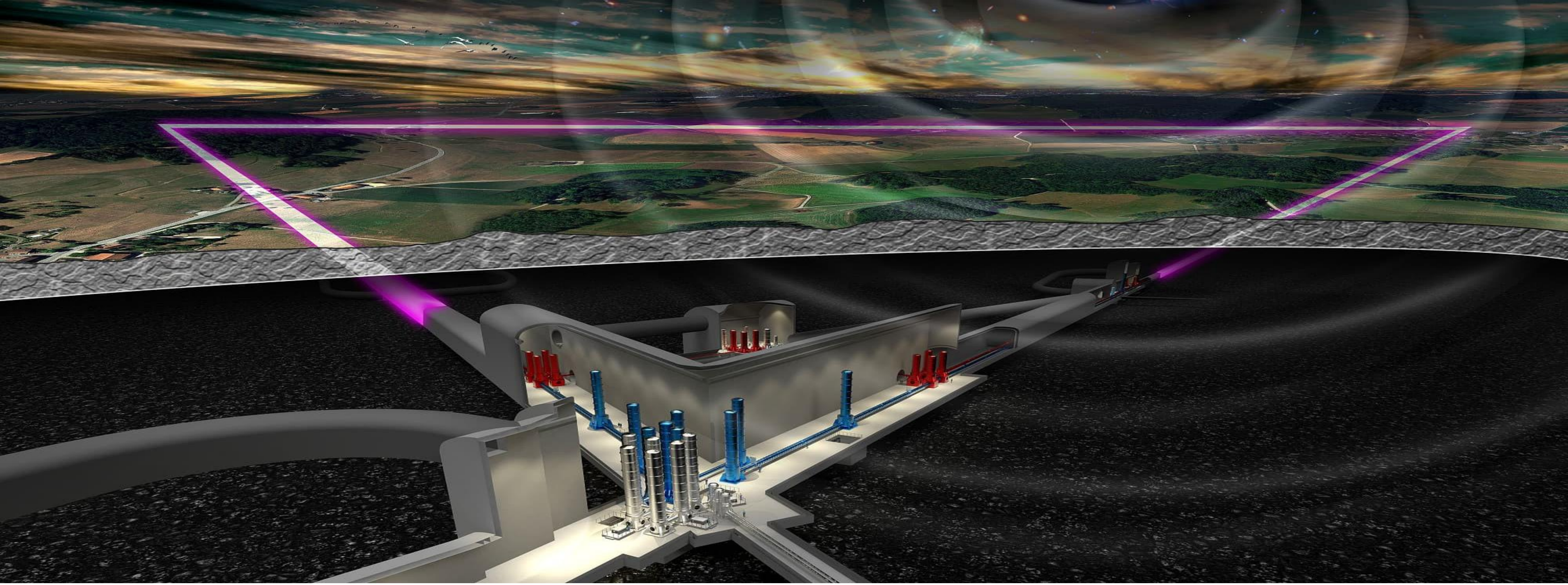


# Needed mitigation factors in frequency domain (average over single day – BH2)



# Needed mitigation factors in frequency domain (average over single day – BHZ)





# Backup

# From seismic displacement to mirror acceleration (in time domain)

- All incoming waves are plane waves (of frequency in range approx. 1-10 Hz) $\vec{\xi} = \vec{\xi}_P + \vec{\xi}_S$
- mirror in **spherical cavern** located in a **quasi-infinite homogenous medium** of density $\rho_0$
- $\boldsymbol{\delta\vec{a}(\vec{r}_0, t)}$ **: acceleration of the mirror** → integrate to strain noise $\frac{\Delta L}{L}$
- $\boldsymbol{\vec{\xi}(\vec{r}_0, t)}$ **: seismic displacement** measured at the cavern wall (close to mirror)

$$\delta\vec{a}(\vec{r}_0, t) = \frac{4\pi}{3} G\rho_0 \left( (+2) \cdot \vec{\xi}_P(\vec{r}_0, t) + (-1) \cdot \vec{\xi}_S(\vec{r}_0, t) \right)$$

$$\boxed{\frac{4\pi}{3} G\rho_0 \approx 8 \cdot 10^{-16} \frac{\mathrm{m}}{\mathrm{s}^2\ \mathrm{nm}}}$$

REMINDER: This is Harms model used for the ET ISB budget!

# ET-NoiseBudget (ISB)

[ET / ISB / Interferometer / ET-NoiseBudget · GitLab](https://gitlab.et-gw.eu/et/isb/interferometer/ET-NoiseBudget)
https://gitlab.et-gw.eu/et/isb/interferometer/ET-NoiseBudget

```python
def spectrum_bodywave(f,Seismic):
    dataSosEnattos = np.loadtxt("acoustic_spectra/bodywave_spectrum_SosEnattos.txt")
    dataTerziet = np.loadtxt("acoustic_spectra/bodywave_spectrum_Terziet.txt")

    if Seismic.Site=='ET':
        bodywave=(5 * gwinc.noise.seismic.seismic_ground_NLNM(f))**2
    elif Seismic.Site =='SosEnattos':
        bodywave=loginterp(dataSosEnattos.T[0],dataSosEnattos.T[3],f)
    elif Seismic.Site =='Terziet':
        bodywave=loginterp(dataTerziet.T[0],dataTerziet.T[3],f)

    return bodywave
```

$$\text{PSD:} \quad S^h_{\text{bw}} = \left(\frac{4\pi}{3} G\rho_0\right)^2 (3p+1) S_{\text{bw}}(\xi_{\text{TM}}, f) \frac{4}{L^2(2\pi f)^4}$$

- bodywave_spectrum_SosEnattos.txt
- bodywave_spectrum_SosEnattos_V.txt
- bodywave_spectrum_SosEnattos_old.txt
- bodywave_spectrum_Terziet.txt
- bodywave_spectrum_Terziet_V.txt
- bodywave_spectrum_Terziet_old.txt

```python
################################################################################
## Noise sources
## All equation numbers refer to those in https://arxiv.org/abs/2003.03434
################################################################################
def body_wave(f,Seismic):
    p = 0.33  # Fraction of body wave spectral density caused by compressional waves
    rock_density = Seismic.RhoR  # kg / m^3
    Sh =  (4/3 * np.pi * constants.G * rock_density)**2 * (3*p + 1) * spectrum_bodywave(f,Seismic) * 4 / (2*np.pi*f)**4  # Equation 7
    return np.sqrt(Sh)

def rayleigh_wave(f,Seismic):
    vr = spectrum_rayleigh_dispersion(f,Seismic)
    vs = 1.1 * vr  # Shear wave dispersion TODO: pulled from slide 16 of http://rses.anu.edu.au/~nick/teachdoc/lecture5.pdf, find better source
    vp = 2 * vr  # TODO: quick guess
    kr = 2 * np.pi * f / vr

    qp = 2 * np.pi * f * np.sqrt(1 / vr**2 - 1 / vp**2)
    qs = 2 * np.pi * f * np.sqrt(1 / vr**2 - 1 / vs**2)
    zeta = np.sqrt(qp / qs)

    h = -Seismic.Height  # Detector depth in m
    gamma = 0.8  # Factor quantifying cancellation of newtonian noise
    density_surface = Seismic.RhoS  # Density of surface in kg / m^3

    r0 = kr * (1 - zeta)  # eq. 3
    sh = -kr * (1 + zeta) * np.exp(-kr * h)  # eq. 4
    bh = 2/3 * (2 * kr * np.exp(-qp * h) + zeta * qs * np.exp(-qs * h))  # eq. 5
    R = np.abs((sh + bh) / r0)**2  # eq. 6
    SR = (2 * np.pi / np.sqrt(2) * gamma * constants.G * density_surface)**2 * R * spectrum_rayleigh_vertical(f,Seismic) * 4 / (2 * np.pi * f)**4

    return np.sqrt(SR)
```