Advanced Virgo

-

Advanced Virgo sensitivity curve study

VIR-0073B-12

Authors:

M Punturo[1]

Issue: C

Date: July 24, 2012

[1] INFN - Sezione di Perugia, Italy,

Abstract

This document describes the sensitivity curve modeling of the Advanced Virgo detector.

Table 1: Document history

Issue	**Date**	**Description**
A	09-03-2012	First description of the fundamental noises, as currently implemented in gwinc. Just started the computation of the AdV noise and missing any evaluation of technical noises (to be inserted in a future release).
B	29-03-2012	Residual seismic noise introduced in gwinc and in the note. Losses budget introduced in the quantum noise chapter
C	24-07-2012	Detection distance corrected by Patrick Sutton, because of an incorrect integration constant, resulting in to an increment of about 6% in the horizon (function int73.m). Reference configuration file modified reducing the clear aperture for the clipping losses to 33 cm dominated by the baffles and not by the coating. Suspension thermal noise still to be modified because computed taking in account a reference mass anymore present in the AdV design.

Contents

1 Seismic Noise

The seismic noise, consisting in to the natural and anthropogenic shaking of soil, enters in the noise budget of AdV through both the residual vibration transmitted through the SA to the mirrors and the direct coupling due to the Newtonian attraction force of the suspended test masses to the soil (the so–called Newtonian or Gravity Gradient Noise [1–3]).

Seismic activity at the Virgo site is obviously affected by the weather conditions (wind, sea excitation) and by the human activity (for few examples see [4] and [5]). In particular, at the Virgo site, the wind effect is visible under 0.1 Hz, the ocean between 0.1 and 0.2 Hz, the Tirreno sea between 0.2 and 1 Hz, the traffic between 1 and 5 Hz [6]; this generate a *structured* noise, mainly below the detection cut–in frequency. This noise (so called μ–seism) obviously fluctuates according to the environmental condition, as shown in Fig.1 and in Fig.2.

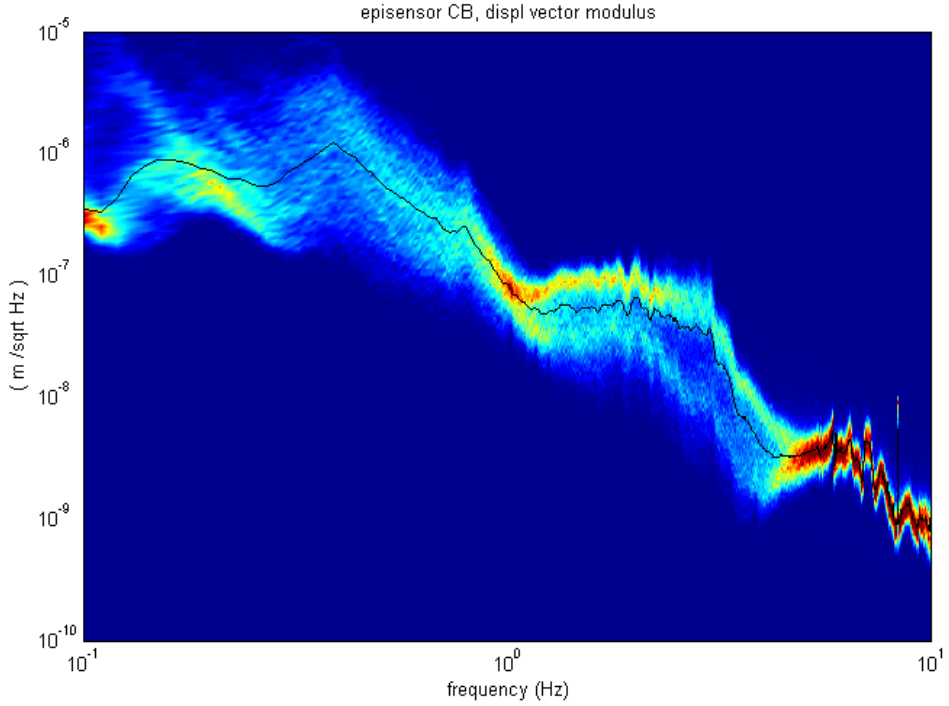

Figure 1: Spectral amplitude of the (μ–) seismic displacement noise measured in the Virgo Central Building (CB). The color level indicates the persistence of the noise amplitude in the correspondent range [7].

An accurate simulation of the seismic noise should take in account all these structures; this has been attempted trying to reproduce the main bumps at low frequency and the $\approx 10^{-7}/f^2$ behavior at high frequency

$$x_{\rm seism} = a_1 \cdot \left| \prod_{j=1,3} \left(f^2 - f_j^2 + i\frac{f_j f}{Q_j} \right)^{p_j} \right| + a_2 \cdot \sum \text{bumps} \qquad (1)$$

Bump functions are coded inside groundSeism.m (see Appendix C). In gwinc these parameters are contained in **ifo.Seismic**.

In Fig.3 the seismic noise measured in the Virgo CB is compared with the model of Eq.1 in case of quiet μ–seism. In Fig.4 the same comparison is made in case of noisy environmental conditions.

1.1 The Virgo seismic filtering system

The most recent description of the filtering performance of the Virgo Super–Attenuator (SA) is in [8]. Essentially some reduction of the filtering capability has been seen, probably due to a magnetic short–cut in the vertical

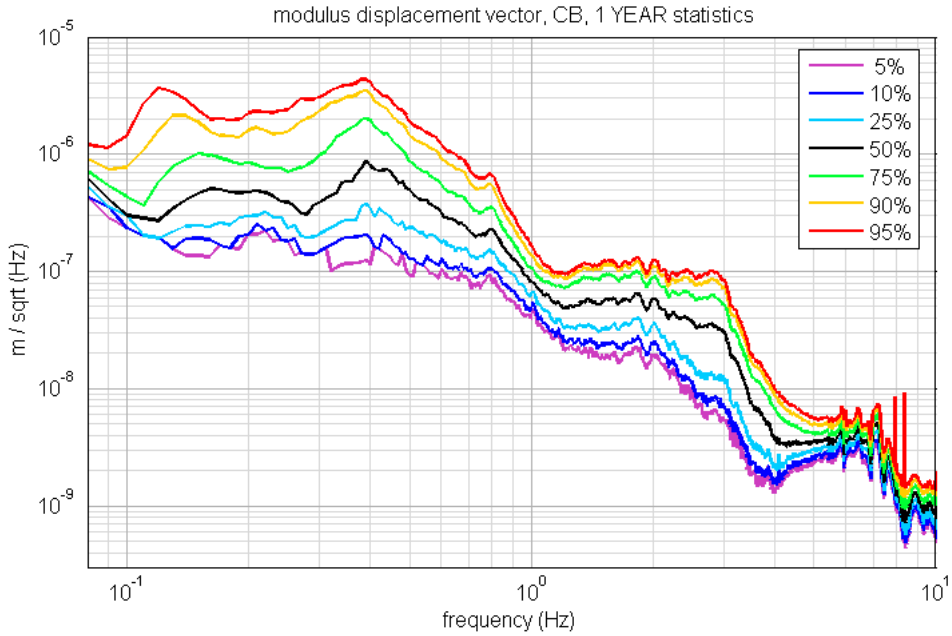

Figure 2: Statistical property of the spectral amplitude of the (μ–)seismic displacement noise measured in the Virgo Central Building (CB). For each frequency bin, it is reconstructed the probability to find a seismic amplitude below the threshold indicated by each curve [7].

chain of filters, but it is still valid to approximate the transfer function (TF) of the whole chain, by the product of the TFs of each stage. A detailed description of the SA TF computation is in the Paolo Ruggi Thesis [9] and here only the results are reported. In Fig.5 the Horizontal and Vertical TFs are shown [6]. The *Blue curve* represents the TF from the filter 0 to the mirror for the horizontal displacement; the determine the effective filtering the inverted pendulum filtering should be considered. For technical reason, it is preferred to measure the seismic noise on the top of the IP and to consider the filtering properties of the chain F0–mirror. The *Red curve* represents the Vertical TF ground–to–mirror. It is clear that in the in–band frequency range, the role of the vertical–to–horizontal coupling is crucial to determine the effective filtering of the SA. A mechanical coupling angle of about $\theta_c \simeq 10^{-3}$ is expected. Adopting such as coupling angle it is possible to compute the residual seismic noise at the level of the mirror, as shown in Fig.6.

1.2 Gravity Gradient Noise

The Gravity gradient noise or Newtonian Noise [1–3] is due to the direct coupling, through the Newtonian attraction force, of the suspended test masses to the soil. The latest and more general evaluation of this noise source has been done by G. Cella in [10] (see Eq.50 in [10]); for low frequencies and surface detector that evaluation coincides with the previous estimation [2] of the spectral amplitude (in terms of h $1/\sqrt{\text{Hz}}$):

$$h_{GGN}(\omega) = \frac{2\sqrt{\pi}}{3}\frac{4\pi}{L}\left(\frac{\omega_{GG}}{\omega}\right)^2 \cdot x_{seism}(\omega) \;\; ; \;\; \omega_{GG} = \sqrt{G\rho_{soil}} \tag{2}$$

where G is the (universal) gravity constant and ρ_{soil} is the averaged density of the soil in the AdV site ($\rho \simeq 2300 - 2700\,[\text{kg/m}^3]$ [11, 12]).

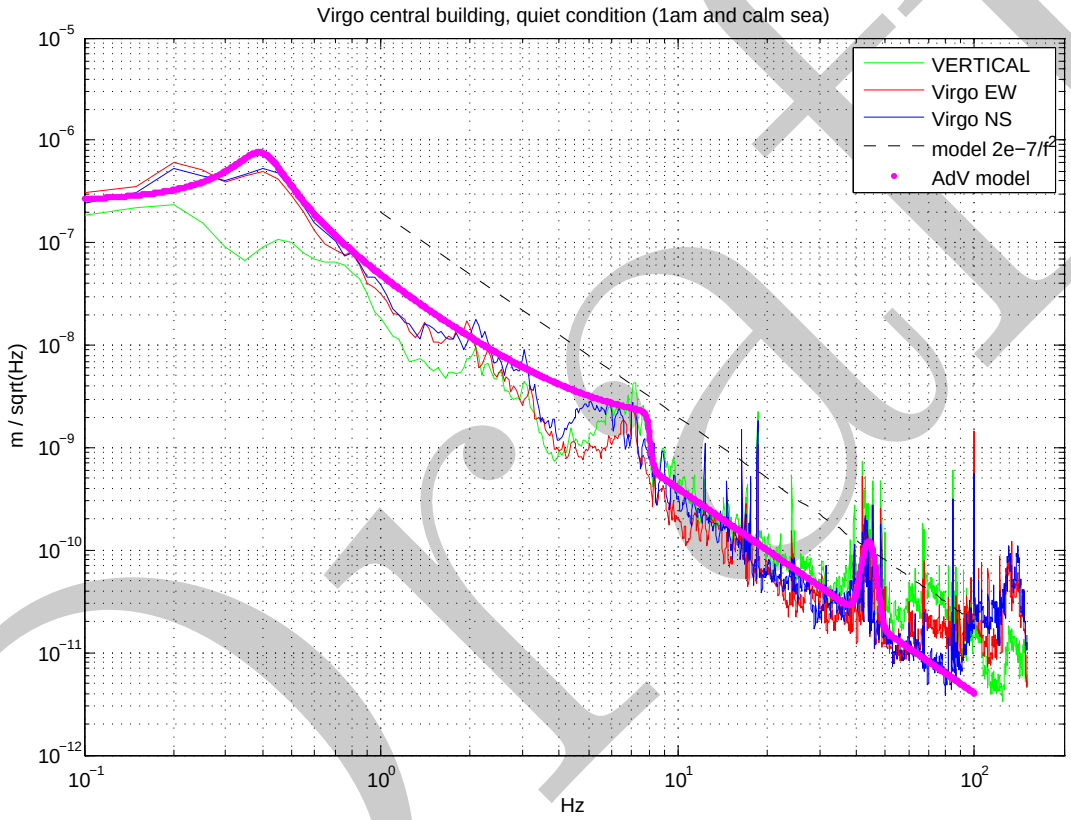

Figure 3: Seismic noise in the Virgo CB [7], in a condition of **quiet** μ–seism. The green curve shows the spectral amplitude of the vertical seismic red curve the East–West horizontal vibration and the blue curve the North–South horizontal seismic vibration. The dashed black line is the $10^{-7}/f^2$ simple model. The magenta markers shows the prediction of Eq.1, computed with the low noise parameters.

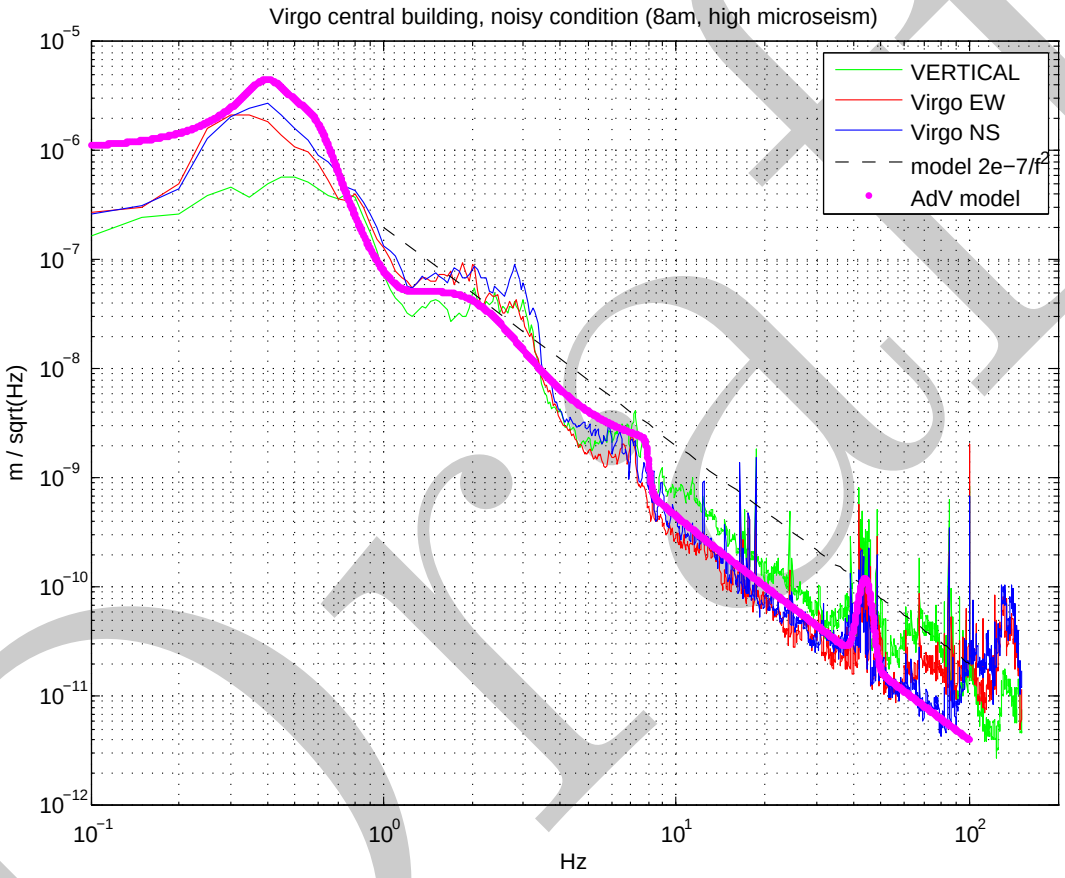

Figure 4: Seismic noise in the Virgo CB [7], in a condition of **noisy** μ–seism. The green curve shows the spectral amplitude of the vertical seismic red curve the East–West horizontal vibration and the blue curve the North–South horizontal seismic vibration. The dashed black line is the $10^{-7}/f^2$ simple model. The magenta markers shows the prediction of Eq.1, computed with the high noise parameters.

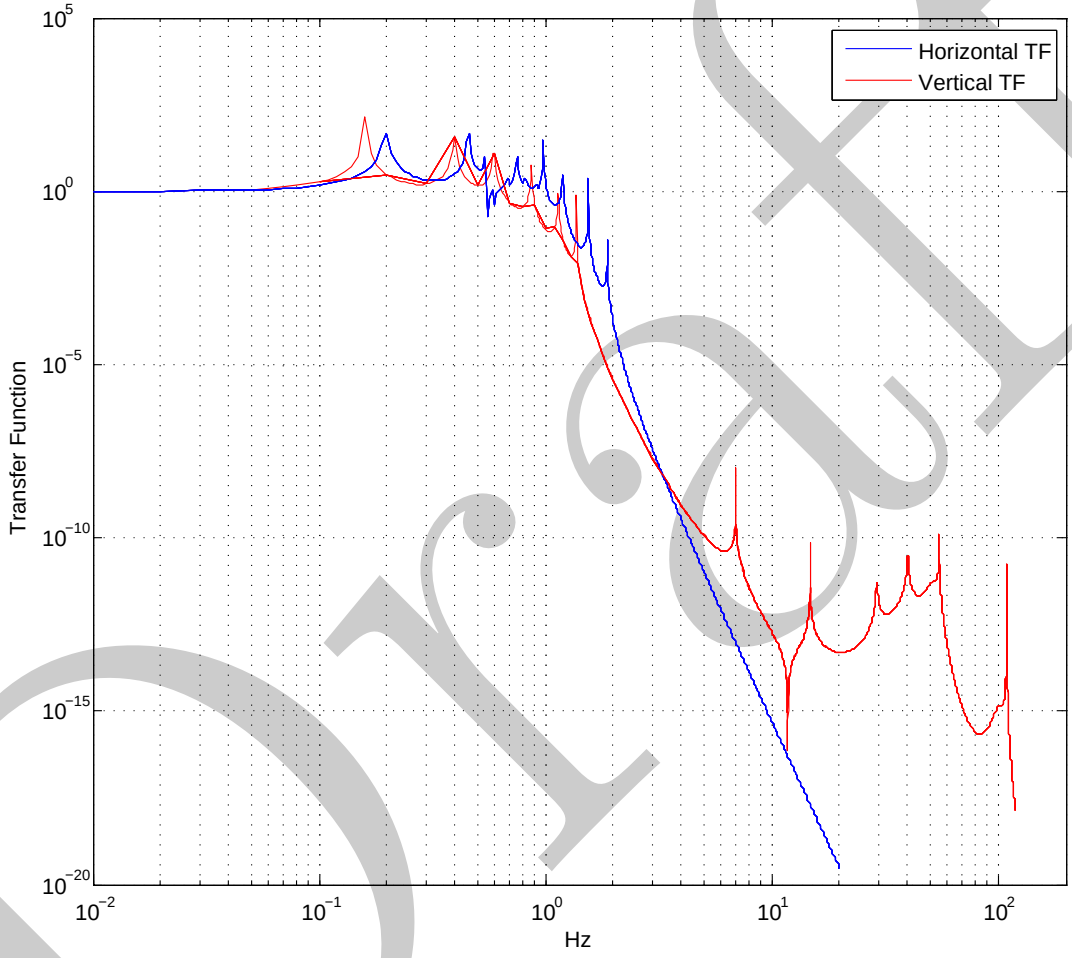

Figure 5: Transfer Functions (TF) of the Virgo SA [6]: *Blue curve*, Horizontal TF from the filter 0 to the mirror. To determine the effective filtering the role of the inverted pendulum (IP) must be inserted. *Red curve*, Vertical transfer function from ground to mirror (in this plot two curves, relative to a different frequency sampling, have been overlapped).

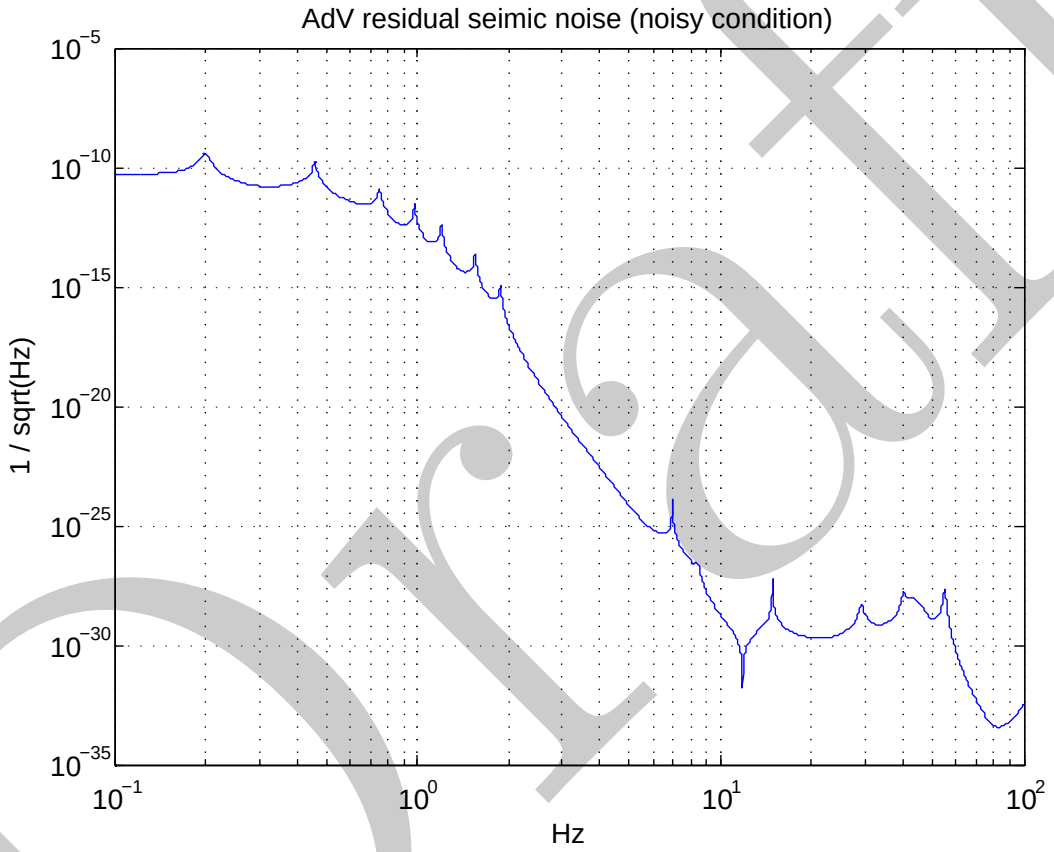

Figure 6: Residual seismic noise [6] at the level of the mirror (dimensionless strain unit), computed using a TFs similar to the one in Fig.5, a vertical–to–horizontal coupling angle coupling angle $\theta_c \simeq 10^{-3}$ and the input seismic noises of Eq.1.

2 Test mass Thermal Noises

Four thermal noise contribution can be identified in the test masses of the interferometer:

- Classical Brownian motion of the mirror
- Thermo–Dynamical fluctuation of the mirror bulk
- Thermo–Dynamical fluctuation of the mirror coating
- Thermo–refractive fluctuation of the mirror coating

2.1 Classical Brownian motion of the mirror

The evaluation of the Brownian fluctuation of the AdV mirror can be evaluated through the direct application of the fluctuation–dissipation theorem [13], as shown by Levin [14] and Bondu et. al. [15]. The displacement power spectrum at low frequency (well below the first resonant mode) of a single mirror is:

$$\langle x_{Brown}(\omega)\rangle^2 = \frac{8k_BT}{\omega}\phi_{mirr}(f)\cdot U \tag{3}$$

where ϕ_{mirr} is the effective loss angle of the mirror and U the the elastic energy stored in the mirror under a normalized gaussian pressure having the shape of the laser beam. The energy U has been evaluated by Levin:

$$U = \frac{1-\sigma_{FS}^2}{2\sqrt{\pi}\cdot E_{FS}\cdot w_{beam}} \tag{4}$$

where w_{beam} is the size of the beam illuminating the mirror. The formula 4 is valid for an infinite size mirror and it has been evaluated in the case of a finite size mirror (thickness h_m, radius R_m) by Vinet [16]:

$$U = U_0 + \Delta U \tag{5}$$

where

$$U_0 = \frac{1-\sigma_{FS}^2}{\pi R_m\cdot E_{FS}}\sum_{k=0}^{\infty} W_k \frac{\exp\left[-\left(\xi_k\frac{w_{beam}}{2R_m}\right)^2\right]}{\xi_k J_0(\xi_k)^2} \tag{6}$$

$$W_k = \frac{1-exp\left(-4\xi_k\frac{h_m}{R_m}\right)+4\left(\xi_k\frac{h_m}{R_m}\right)\cdot exp\left(-2\xi_k\frac{h_m}{R_m}\right)}{\left[1-exp\left(-2\xi_k\frac{h_m}{R_m}\right)\right]^2-4\left(\xi_k\frac{h_m}{R_m}\right)^2 exp\left(-2\xi_k\frac{h_m}{R_m}\right)}$$

where ξ_k are the zeros of the Bessel function J_1. The finite size mirror correction ΔU is

$$\Delta U = \frac{R_m^2}{6\pi h_m^3\cdot E_{FS}}\left[\left(\frac{h_m}{R_m}\right)^4 + 12\sigma_{FS}\cdot\Sigma\cdot\left(\frac{h_m}{R_m}\right)^2 + 72\left(1-\sigma_{FS}\right)\Sigma\right] \tag{7}$$

$$\Sigma = \sum_{k=0}^{\infty}\frac{exp\left[-0.5\left(\xi_k\frac{w_{beam}}{2R_m}\right)^2\right]}{\xi_k^2 J_0(\xi_k)}$$

These formulas are implemented in the code shown in the Appendices B and D.

2.1.1 The coating contribution

The effective loss angle ϕ_{mirr} should take in account the substrate dissipation ϕ_{bulk} and all the contributions given by parassitic losses (surface, coatings,...); in particular the coating contribution can be modeled [17] as

$$\phi_{mirr} = \phi_{bulk}(f) + d_{coat}\frac{\delta U}{U}\phi_{coat} \tag{8}$$

where δU is the energy density at the surface integrated over the surface and d_{coat} is the thickness of the coating. The factor $\delta U/U\phi_{coat}$ can be divided in the terms parallel ($\|$) and perpendicular ($\perp$) to the optic face:

$$\frac{\delta U}{U}\phi_{coat} = \left[\frac{\delta U_\perp}{U}\right]\cdot\phi_\perp + \left[\frac{\delta U_{\|-\perp}}{U}\right]\cdot\left(\phi_\| - \phi_\perp\right) + \left[\frac{\delta U_\|}{U}\right]\cdot\phi_\| = \tag{9}$$
$$= \frac{E_{FS}}{1-\sigma_{FS}^2}\frac{1}{\sqrt{\pi}w_{beam}}\cdot$$
$$\left\{\frac{1}{E_\perp}\left[1-\frac{2\sigma_\perp^2 E_\|}{E_\perp\left(1-\sigma_\|\right)}\right]\phi_\perp + \frac{E_\|}{E_\perp E_{FS}}\frac{\sigma_\perp\left(1+\sigma_{FS}\right)\left(1-2\sigma_{FS}\right)}{\left(1-\sigma_\|\right)}\left(\phi_\| - \phi_\perp\right) + \frac{E_\|}{E_{FS}^2}\frac{\left(1+\sigma_{FS}\right)^2\left(1-2\sigma_{FS}\right)^2}{1+\sigma_\|^2}\phi_\|\right\}$$

where the loss angles and Young's modulus perpendicular and parallel to the coating plane are defined by

$$\phi_\| = \frac{E_{n1}\phi_{n1}d_{n1} + E_{n2}\phi_{n2}d_{n2}}{E_\| d_{coat}} \tag{10}$$
$$\phi_\perp = \frac{E_\perp}{d_{coat}}\left(\frac{\phi_{n1}d_{n1}}{E_{n1}} + \frac{\phi_{n2}d_{n2}}{E_{n2}}\right)$$
$$E_\| = \frac{E_{n1}d_{n1} + E_{n2}d_{n2}}{d_{coat}}$$
$$E_\perp = \frac{d_{coat}}{\frac{d_{n1}}{E_{n1}} + \frac{d_{n2}}{E_{n2}}}$$

The two Poisson's ratios σ_1 and σ_2 are defined by

$$\sigma_1 \simeq \frac{\sigma_{n1} + \sigma_{n2}}{2} \tag{11}$$
$$\sigma_2 = \frac{\sigma_{n1}E_{n1}d_{n1} + \sigma_{n2}E_{n2}d_{n2}}{E_{n1}d_{n1} + E_{n2}d_{n2}}$$

Here $d_{coat} = d_{n1} + d_{n2}$ is the coating thickness. The index $n1$ indicates the high refraction index coating material (Ta_2O_5) and $n2$ the low refraction index coating material (SiO_2).

The coating Brownian noise is computed in the Appendices E and F.

2.1.2 Substrate loss angle

Penn and coll. [18] shown that the loss angle of fused silica samples (of volume V and surface S) measured in different experiments by the GW community follows a model that takes in account three dissipation mechanisms: the surface (ϕ_{surf}), the bulk ($\phi_{bulk}(f)$) and the thermo-elastic dissipation (ϕ_{th}). Suprasil loss angle can be modeled as:

$$\phi_{sub}\left(f, \frac{V}{S}\right) = \phi_{surf} + \phi_{bulk}(f) + \phi_{th} = \tag{12}$$
$$= C_1\left(\frac{V}{S}\right)^{-1} + C_2\left(\frac{f}{1Hz}\right)^{C_3} + C_4\phi_{th}$$

The coefficients C_2 and C_3 are in gwinc respectively given by the variables **ifo.Materials.Substrate.c2** and **ifo.Materials.Substrate.MechanicalLossExponent** (see A) and the computation is performed in Appendix B.

2.1.3 Surface losses contribution

In gwinc (see subbrownian.m in Appendix B) is computed an additional dissipation on the Brownian noise given by surface losses and computed as:

```
csurfETM = alphas*(1-2*sigma)/((1-sigma)*Y*pi*wETM^2);
csurfITM = alphas*(1-2*sigma)/((1-sigma)*Y*pi*wITM^2);
csurf = 8 * kBT * (csurfITM + csurfETM) ./ (2 * pi * f);
% account for 2 ITM and 2 ETM, and convert to strain whith 1/L^2
n = 2 * (csurf + cbulk) / L^2;
```

Currently is not clear to me the correctness of this evluation, because the mirror, essentially, has not free surface. Hence, either only the coating contribution must be accounted (considering also the AR coating dissipation) or I would use the modeling in formula (13) of [17].

2.2 Substrate thermo-elastic noise

The modeling of the substrate thermo-elastic noise is fully described in [19]. The power spectrum of the single mirror displacement can be modeled as:

$$S^{FTM}(\omega) = C^2_{FTM} S^{ITM}(\omega) \tag{13}$$

where the superscripts stand for ITM - Infinite Test Mass and FTM - Finite Test Mass (hence C^{FTM} is a correction factor quite close to the unity). The thermo-elastic noise for an infinite test mass is

$$S^{ITM}(\omega) = \frac{8\left(1+\sigma_{FS}\right)^2 \kappa_{FS} \alpha^2_{FS} k_B T^2}{\sqrt{2\pi} C^2_V \rho^2_{FS} \left(w_{beam}/\sqrt{2}\right)^3 \omega^2} \tag{14}$$

where κ_{FS} is the thermal conductivity of the fused silica, α_{FS} is the linear thermal expansion coefficient, ρ_{FS} the density and C_V is the specific heat per unit mass at constant volume. The infinite test mass thermo-elastic contribution is computed in Appendix G.

The correction factor is (formula (46) of [19])

$$C^2_{FTM} = \frac{\left(\sqrt{\pi} w_{beam}\right)^3}{R^3_m} \times \left\{ \frac{R^5_m h_m \tilde{c}^2_1}{(1+\sigma)^2} + \sum_{n=1}^{\infty} \frac{R^5_m k_n p^2_n \left(1-Q_n\right) J^2_0(\xi_n)}{\left[\left(1-Q_n\right)^2 - 4h_m k^2_n Q_n\right]^2} \times \left[\left(1-Q_n\right)^2\left(1+Q_n\right) + 8h_m k_n Q_n \left(1-Q_n\right) + 4h^2_m k^2_n \left(1+Q_n\right)\right] \right\} \tag{15}$$

where

$$k_n = \xi_n / R_m \tag{16a}$$

$$Q_n = \exp\left(-2k_n h_m\right) \tag{16b}$$

$$p_n = \frac{2}{R^2_m J^2_0\left(\xi_m\right)} \int_0^{R_m} \frac{\exp\left(-r^2/r^2_0\right)}{\pi r^2_0} J_0\left(k_n r\right) r dr \simeq \frac{\exp\left(-k^2_n r^2_0/4\right)}{\pi R^2_m J^2_0\left(\xi_n\right)} \tag{16c}$$

$$r_0 = w_{beam}/\sqrt{2} \tag{16d}$$

where the approximation in Eq.16c stands for small values of n (see *caveat* in page 5 of [19]). The finite test mass correction to the thermo-elastic noise is computed in Appendix H.

2.3 Thermo-optical noises

Thanks to the Fluctuation-Dissipation theorem [13], the thermal dissipation in a material can be linked to the temperature fluctuation. In an "optical material" this can be related to an optical phase noise through two different couplings: the linear thermal expansion coefficient α (defined as thermo-elastic noise) and the refraction index themperature gradiend $\beta = dn/dT$ (defined as thermo-refractive noise). Several authors have treated the two components separately. For example the thermo-elastic noise in the coating has been described in [20–22] (as guided reading I suggest the Einstein Telescope internal note [23]); the corresponding power spectrum of the displacement noise induced by thermo-elastic fluctuation in the coating is expressed, for example, by formula (7) in [22]. Thermo-refractive noise is instead separately described in [24] (see, in that paper, for example formula (9)). An complete view of the thermo-dynamical and thermo-refractive noises, unified under the name thermo-optical, has been performed by M. Evans and collaborators in [25] and hereafter I will use their results.

Thermo-optical noise is driven by the thermal fluctuation in the coating; the power spectrum of the coating thermal fluctuation,investigated by a Gaussian beam, is

$$S_{TO}^{\Delta T} = \frac{2\sqrt{2}}{\pi} \frac{k_B T^2}{w_{beam}^2 \sqrt{\kappa_{FS} C_{FS} \omega}} \tag{17}$$

This thermal fluctuation determines a displacement noise through the thermal gradient:

$$\frac{\partial \Delta z}{\partial T} = \frac{\partial \Delta z_{TE}}{\partial T} + \frac{\partial \Delta z_{TR}}{\partial T} = \frac{\partial \Delta z}{\partial \phi} \frac{\partial \phi}{\partial T} = \partial_\phi^z \frac{\partial \phi}{\partial T} \tag{18a}$$

$$\frac{\partial \Delta z}{\partial \phi} = \partial_\phi^z = -\frac{\lambda}{4\pi} \tag{18b}$$

where the suffices "TE" and "TR" stand for "Thermo-Elastic" and "Thermo-Refractive" (in this subsection we will use also the suffices "s" and "c" to indicates "substrate" and "coating" related properties. Furthermore, we are here adopting the same formalism described in [25]: a bar used above a symbol indicates an "effective" coefficient, having the same unit and meaning of its barless counterpart, but averaged in a context o a semi–infinite medium). The full evaluation is performed in [25] and the final formula is

$$S_{TO}^{\Delta z} = S_{TO}^{\Delta T} \times \left(\partial_\phi^z \frac{\partial \phi}{\partial T} - \overline{\alpha}_s d \frac{C_c}{C_s} \right)^2 \tag{19}$$

Let first focus on the terms within the parenthesis in Eq.19; the second term subtracts the thermo-elastic contribution of the substrate whereas the first term contains both the thermo-elastic and thermo-refractive contributions of the coating. If we indicate with the index "0" the input medium (the "vacuum") and "N" is the number of layers of the dielectric coating) we can write

$$\frac{\partial \phi_c}{\partial T} = \sum_{k=0}^{N} \frac{\partial \phi_c}{\partial \phi_k} \frac{\partial \phi_k}{\partial T} \tag{20}$$

where the first factor describes the overall reflection phase variation due to the round–trip phase change in each layer and the second factor describes the phase gradiend of each layer with the temperature. The computation of these terms is quite tricky and all the explanations can be found the in Appendix B in [25]; here we make an oversimplified summary. The reflectivity between the layer 1 and 2 of a coating is

$$r_{1,2} = \frac{n_1 - n_2}{n_1 + n_2} \tag{21}$$

If we pile–up two transitions

$$r_{1,2,3} = \frac{-r_{2,1} + r_{2,3} e^{-i\phi_2}}{1 - r_{2,1} r_{2,3} e^{-i\phi_2}} \tag{22}$$

where ϕ_2 is the round-trip phase in layer 2. Noting that $r_{k,k+1} = -r_{k+1,k}$ and defining $r_k \equiv r_{k,k+1}$ it is possible to write the recursive formula for the effective reflectivity of a coating layer,including the round-trip in that layer:

$$\overline{r}_k \equiv e^{-i\phi_k} r_{k,k+1,\ldots,N} = e^{-i\phi_k} \frac{r_k + \overline{r}_{k+1}}{1 + r_k \overline{r}_{k+1}} \tag{23}$$

The overall reflection of the coating is then $r_c = \overline{r}_0$ and its phase $\phi_0 = \Delta_c / \partial_\phi^z$ takes in account the total change of thickness Δ_c of the coating. Now it is possible to compute the first factor of the Eq.20:

$$\frac{\partial \phi_c}{\partial \phi_k} = \Im\left(\frac{1}{\overline{r}_0} \frac{\partial \overline{r}_0}{\partial \phi_k}\right) \tag{24a}$$

$$\frac{\partial \overline{r}_k}{\partial \phi_j} = \begin{cases} e^{-i\phi_k} \frac{1-r_k^2}{(1+r_k \overline{r}_{k+1})^2} \frac{\partial \overline{r}_{k+1}}{\partial \phi_j} & \text{if } k < j \\ -i\overline{r}_k & \text{if } k > j \\ 0 & \text{if } k = j \end{cases} \tag{24b}$$

Finally, it is possible to compute the second factor of Eq.20

$$\frac{\partial \phi_k}{\partial T}|_{k>0} = \frac{4\pi}{\lambda} \left(\beta_k + \overline{\alpha}_k n_k\right) d_k = \frac{4\pi}{\lambda} B_k d_k \tag{25a}$$

$$\frac{\partial \phi_0}{\partial T} = -\frac{4\pi}{\lambda} \sum_{k=1}^{N} \overline{\alpha}_k d_k = \overline{\alpha}_c \frac{d}{\partial_\phi^z} \tag{25b}$$

Eq.25b represents the geometric variation of the whole coating, hence the Thermo-elastic contribution. Eq.25a is related to the optical path variation, hence it represents the thermo-refractive contribution. In the Eq.25a and 25b have been used the following definitions

$$d = \sum_{k=1}^{N} d_k \tag{26a}$$

$$\overline{\alpha}_k = \alpha_k \frac{1+\sigma_s}{1-\sigma_k} \left[\frac{1+\sigma_k}{1+\sigma_s} + (1-2\sigma_s) \frac{E_k}{E_s}\right] \tag{26b}$$

$$\overline{\alpha}_c = \sum_{k=1}^{N} \overline{\alpha}_k \frac{d_k}{d} \tag{26c}$$

$$B_k \equiv \beta_k + \overline{\alpha}_k n_k \tag{26d}$$

where d is the total thickness of the coating, α_k is the thermal expansion coefficient for a given layer k and $\overline{\alpha}_c$ is the effective linear expansion coefficient of the whole coating. For sake of completness hereafter are shown the latest definition needed to perform the above computations:

$$\overline{C}_c = \sum_{k=1}^{N} C_k \frac{d_k}{d} \tag{27a}$$

$$\overline{\kappa}_c = \left(\sum_{k=1}^{N} \frac{1}{\kappa_k} \frac{d_k}{d}\right)^{-1} \tag{27b}$$

$$\overline{\alpha}_s \simeq 2\alpha_s \left(1 + \sigma_s\right) \tag{27c}$$

where $\overline{\alpha}_s$ is the effective expansion coefficient of the substrate in a semi-infinite mirror. In gwinc this tricky computation is performed in getCoatTOPhase.m (see Appendix I).

The previous computations have been performed under the hypothesis that the coating thickness is quite smaller than the thermal diffusion lenght r_T

$$r_T = \sqrt{\frac{\kappa}{C\omega}} \tag{28}$$

Removing this hypothesis and considering also a corrective factor due to the finite size of the mirror, Eq.19 becomes

$$S_{TO}^{\Delta z} \simeq S_{TO}^{\Delta T} \times \Gamma_{tc} \times \left(\Delta\overline{\alpha}_{fsm} d - \overline{\beta}\lambda\right)^2 \tag{29}$$

where Γ_{tc} is a correction factor related to the coating thickness, the first term within the parentesis describes the thermo-elastic contribution and the second term the thermo-refractive one, with the definition

$$\frac{\partial z_{TR}}{\partial T} \equiv -\overline{\beta}\lambda \tag{30}$$

The coating thickness correction factor is

$$\Gamma_{tc} = \frac{p_E^2\Gamma_0 + p_E p_R \varsigma \Gamma_1 + p_R^2 \varsigma^2 \Gamma_2}{R\varsigma^2\Gamma_D} \tag{31}$$

where

$$\Gamma_0 = 2\left(\sinh(\varsigma) - \sin(\varsigma)\right) + 2R\left(\cosh(\varsigma) - \cos(\varsigma)\right) \tag{32a}$$

$$\Gamma_1 = 8\sin(\varsigma/2)\left(R\cosh(\varsigma/2) + \sinh(\varsigma/2)\right) \tag{32b}$$

$$\Gamma_2 = \left(1+R^2\right)\sinh(\varsigma) + \left(1-R^2\right)\sin(\varsigma) + 2R\cosh(\varsigma) \tag{32c}$$

$$\Gamma_D = \left(1+R^2\right)\cosh(\varsigma) + \left(1-R^2\right)\cos(\varsigma) + 2R\sinh(\varsigma) \tag{32d}$$

$$R = \sqrt{\frac{\kappa_c C_c}{\kappa_s C_s}} \tag{32e}$$

$$p_R = \frac{-\overline{\beta}\lambda}{\Delta\overline{\alpha}d - \overline{\beta}\lambda} \quad , \quad P_E = \frac{\Delta\overline{\alpha}d}{\Delta\overline{\alpha}d - \overline{\beta}\lambda} \tag{32f}$$

The parameter ς is a dimensionless, frequency dependent, scale factor defined by

$$\varsigma = \frac{\sqrt{2}d}{r_{Tc}} = \sqrt{\frac{2\omega C_c}{\kappa_c}}d \tag{33}$$

that compares the overall thickness d of the coating to its thermal diffusion lenght r_T. The parameter ς describes the novelty of this new description of the thermo-optic noise; in fact, the correction factor Γ_{tc} can be approximated as:

$$\Gamma_{tc} \simeq \begin{cases} 1 + \frac{p_E^2 + 3\left(p_R - R^2\right)}{3R}\varsigma - \frac{p_E - 3\left(1-R^2\right)}{6}\varsigma^2 & \text{if } \varsigma \ll 1 \\ \frac{2p_E^2}{R(1+R)\varsigma^2} + \frac{p_R^2}{R} & \text{if } \varsigma \gg 1 \end{cases}$$

Hence the thermo-refractive and thermo-elastic components have a relative negative sign if $d \ll r_{Tc}$ (thin coating), are partially coherent and partially canceling if $d \simeq r_{Tc}$ and sum in quadrature if $d \gg r_{Tc}$ (thick coating). In gwinc the thickness correction is evaluated in getCoatThickCorr.m (see Appendix J)

Let jump back to the understanding of the main thermo-optical noise equation 29. The first term within the parenthesis, representing the thermo-elastic contribution, can be decomposed as:

$$\Delta\overline{\alpha}_{fsm} = C_{fsm}\Delta\overline{\alpha} \tag{34a}$$

$$\Delta\overline{\alpha} = \overline{\alpha}_c - \overline{\alpha}_s\frac{C_c}{C_s} \tag{34b}$$

where C_{fsm} is a correction factor, related to the finite size of the mirror. Its evaluation is done in [21]:

$$C_{fsm} = \frac{r_0\sqrt{2}}{R_m}\sqrt{S_2 + \frac{\left(-1 + \frac{12R_m^2}{h_m^2}S_1\right)^2}{4\left(1-2\sigma_s\right)^2}} \tag{35a}$$

$$S_1 = \sum_{n=1}^{\infty} \frac{\exp\left(\frac{-k_n^2 r_0^2}{4}\right)}{\xi_n^2 J_0(\xi_n)} \tag{35b}$$

$$S_2 = \sum_{n=1}^{\infty} \frac{\exp\left(\frac{-k_n^2 r_0^2}{2}\right)}{J_0^2(\xi_n)} \left[\frac{(1-Q_n)^2}{(1-Q_n)^2 - 4k_m^2 h_m^2 Q_n}\right]^2 \tag{35c}$$

(see Eq.16a and successive for the definition of k_n, Q_n and r_0). In gwing the finite mirror size coating correction is computed in getCoatFiniteCorr.m (see Appendix K).

It is useful to note that, in high reflectors, composed by quarter–wave doublets of high reflectivity material "H" and low reflectivity material "L" $\overline{\beta}$ is given by

$$\overline{\beta}_{QW} \simeq \frac{n_L^2 B_H + n_H^2 B_L}{4(n_H^2 - n_L^2)} \tag{36}$$

and, if the last layer is a low reflectivity protection layer the previous equation is modified in

$$\overline{\beta} \simeq \frac{\overline{\beta}_{QW}}{n_L^2} + \frac{B_L}{4n_L^2} \tag{37}$$

3 Suspension Thermal Noise

The full description of the thermal noise due to the AdV suspension is performed in [26]; here we summarise the results.

The novelty of the PPP model in [26] is the fact that it takes in account the contributions to the thermal noise due to the dissipations both in the Marionette and in the Reference Mass. Let suppose to indicate with the (1) the Mirror, (2) the Reference Mass and (3) the mirror (as done in Section 1 of [26]; in gwinc, instead, (1) indicates the Marionette and (3) the mirror); these are "coupled harmonic oscillators" and the equations of motion assume a matrix aspect (see Eq.32 in [26]):

$$\widetilde{F} \equiv Z \cdot \dot{\widetilde{X}} \quad \rightarrow \quad Z = \frac{1}{i\omega}\left[M \cdot \left(\Omega + i\omega\Gamma - \omega^2 I\right)\right] \tag{38}$$

where Z is the mechanical impedance 3×3 matrix, M is the diagonal matrix of the masses, I the identity matrix, Ω the matrix of the pendulum (complex)frequencies and Γ is the matrix of the possible viscous dissipations:

$$M = \begin{pmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix} \tag{39a}$$

$$\Omega = \begin{pmatrix} \omega_1^2 & 0 & -\omega_1^2 \\ 0 & \omega_2^2 & -\omega_2^2 \\ -\frac{\mu_1}{\mu_3}\omega_1^2 & -\frac{\mu_2}{\mu_3}\omega_2^2 & \frac{\mu_1\omega_1^2+\mu_2\omega_2^2+\omega_3^2}{\mu_3} \end{pmatrix} \tag{39b}$$

$$\Gamma = \begin{pmatrix} \gamma_1 & 0 & -\gamma_1 \\ 0 & \gamma_2 & -\gamma_2 \\ -\frac{\mu_1}{\mu_3}\gamma_1 & -\frac{\mu_2}{\mu_3}\gamma_2 & \frac{\mu_1\gamma_1+\mu_2\gamma_2+\mu_3\gamma_3}{\mu_3} \end{pmatrix} \tag{39c}$$

$$\omega_j^2 = \omega_{p\,j}^2 + \omega_{w\,j}^2 = \omega_{p\,j}^2\left[(1+D_j) + iD_j\phi_j\right] \ , \ \omega_{p\,j}^2 = \frac{g}{L_j} \tag{39d}$$

The term $\omega_{w\,j}$ is related to the elasticity of the suspension wires; neglecting for simplicity the index j:

$$\omega_w^2 = \omega_p^2 \frac{b}{2L}\sqrt{\frac{nEI}{mg}} \equiv \omega_p^2 D \tag{40}$$

where n_j is the number of suspension wires ($n_1 = n_2 = 4$, $n_3 = 1$), b_j is the number of flexural points ($b_j = 2$), $E$is the Young's modulus of the material composing the wires, $I \equiv (\pi/4)r^4$ is the cross–section momentum (r is the radius of the suspension wire, supposed cylindrical) and D is the so–called Dilution factor. In the present formulation of the mechanical impedance (Eq.38)it has been (a bit artificially) separated the viscous damping processes (γ_j) by the structural dissipation processes, represented by the term ϕ_j in Eq.39d. In the loss angle ϕ_j must be included the bulk structural dissipation of the suspension wire material ϕ_{mat}, the thermo-elastic contribution $\phi_{th}(\omega)$ and the eventual surface losses ϕ_{surf}:

$$\phi_j(\omega) = \phi_{mat\,j} + \phi_{surf\,j} + \phi_{th\,j}(\omega) \tag{41a}$$

$$\phi_{surf} = \phi_{mat}\left(\eta\frac{d_s}{V/S}\right) = \phi_{mat}\frac{4}{r}d_s \tag{41b}$$

$$\phi_{th}(\omega) = \left(\alpha - \beta_E\frac{\Lambda}{E\cdot\pi r^2}\right)^2\frac{ET}{C}\frac{\omega\tau}{1+(\omega\tau)^2}\ ,\ \tau = \frac{C\,(2r)^2}{2.16\cdot 2\pi\cdot\kappa} \tag{41c}$$

where (in Eq. 41b) η is a numerical factor depending on the mode shape ($\eta = 2$ for cylindrical fibres and pendulum mode) d_s is the thickness of the lossy surface, V and S are respectively the volume ($V = 2\pi r_j L_j$) and the lateral surface ($S = \pi r_j^2 L_j$) of the suspension wire. In Eq. 41c Λ is the suspension wire tension ($\Lambda = mg/n$ in case of a mass m suspended by n wires), $C = \rho C_v$ is the heat capacity per volume unit $\left[\mathrm{JK^{-1}m^{-3}}\right]$, κ is the thermal conductivity and $\beta_E \equiv \partial \ln E/\partial T$ (see [27]).

The viscous dissipation coefficients γ_j are related to the mechanical quality factors Q_j of these modes (if the dissipation mechanism is viscous only) by

$$\gamma_j \equiv \frac{\Re(\omega_j)}{Q_j} \tag{42}$$

Now we have all the ingredients to compute the pendulum thermal noise:

$$S_{p,1}(\omega) = \frac{4k_BT}{\omega^2}\Re\left\{\left(Z^{-1}\right)_{11}\right\} = \frac{4k_BT}{\omega^2}\Re\left\{\left[\left[\frac{1}{i\omega}\left[M\cdot\left(\Omega + i\omega\Gamma - \omega^2 I\right)\right]\right]^{-1}\right]_{11}\right\} \tag{43}$$

3.1 Including the violin modes of the mirror suspension wires

In the suspR_AdV.m code of gwinc (see Appendix L), this computation is performed taking in account the results of the section 2 of [26]; the stiffness of the mirror suspension fibres is take in account to evaluate also the violin modes. Eq.38 can be modified in the following way:

$$Z = \frac{1}{i\omega}\left[\mathcal{K} - \omega^2 M\right] \tag{44}$$

including in the $\mathcal{K}$ matrix the role of the Ω and Γ matrices. Obviously $\mathcal{K}$ is the matrix of the elastic constants of the suspension wires:

$$\mathcal{K} = \begin{pmatrix} k_{pend}(\omega) & 0 & -k_{pend}(\omega) \\ 0 & k_2 & -k_2 \\ -k_{pend}(\omega) & -k_2 & k_{pend}(\omega) + k_2 + k_3 \end{pmatrix} \tag{45}$$

where $k_{2,3} = m_{2,3}\left(\omega_{2,3}^2 + i\omega\gamma_{2,3}\right)$. Instead, $k_{pend}(\omega)$ is the full elastic constant of the mirror suspension wires; in case of a simple cylindrical fibre [26, 28]:

$$k_{pend}(\omega) = n_1 EI p_+ p_- \frac{p_+^3\cos(p_-L) + p_+^2 p_-\sin(p_-L) + p_+ p_-^2\cos(p_-L) + p_-^3\sin(p_-L)}{\left(p_+^2 - p_-^2\right)\sin(p_-L) - 2p_+p_-\cos(p_-L)} \tag{46a}$$

$$p_\pm = \sqrt{\frac{\pm\Lambda + \sqrt{\Lambda^2 + 4EI\rho\,(\pi r^2)\,\omega^2}}{2EI}} \tag{46b}$$

Obviously, the violin modes are given by the poles of $k_{pend}(\omega)$ and the pendulum angular frequency can be computed as $\omega_p^2 \simeq \lim_{\omega\to 0}\Re(k_{pend})/m_1$. The "structural" dissipation is hidden in the Young's modulus $E = E_0(1+i\phi)$, whereas the eventual viscous damping can be added *ad hoc* from the measured Q_p as $k_{pend}(\omega) \leftarrow \Re(k_{pend}(\omega)) + i\left[\Im(k_{pend}(\omega)) + \omega/(\omega_p Q_1)\,\Re(k_{pend}(\omega))\right]$

3.1.1 Optimized (Dumbbell–shaped) fibres

In [29] and in the Sec. 2.1 of [26] it is shown that to minimise simultaneously the vertical bouncing frequency (thinner fibres) and the thermo-elastic dissipation (larger fibre diameter, as dictated by Eq.41c) it is possible to adopt the so–called optimised fibres, having larger terminal and a thinner central body (3–segments fibre, $j = 1$ top, $j = 2$ body, $j = 3$ attached to the mirror). It is possible to solve the system of elastic equations that describes the transverse displacement X along the fibre axis y (Eq.49 of [26]) obtaining the three displacement solutions one for each segment of the fibre):

$$X_j = A_j e^{-p_{+,j} - y} + B_j e^{-p_{+,j}(L_j - y)} + C_j \cos(p_{-,j} y) + D_j \sin(p_{-,j} y) \quad , \quad j = 1, 2, 3. \tag{47}$$

The twelve integration constants are determined by the 4 boundary conditions

$$X(0) = 0; \quad X^{(i)}(0) = 0; \quad X(L) = \delta; \quad X^{(i)}(L) = 0; \tag{48}$$

where δ is the displacement of the end terminal of the fibre and the suffix (i) indicates the derivative with respect the longitudinal coordinate y, and by the 8 joining conditions (2 interfaces times (X, $X^{(i)}$, force and moment)). It is the possible to write an effective loss angle weighting the diameter dependent loss angle ϕ (surface and thermo–elastic) with the elastic energy distribution:

$$\phi = \frac{\int_0^L \phi(y)\, u_{\text{elastic}}(y)\, dy}{\int_0^L u_{\text{elastic}}(y)\, dy} \tag{49a}$$

$$u_{\text{elastic}}(y) = \frac{1}{2} EI \left[X^{(ii)}\right]^2 \quad , \quad U_{\text{elastic}} \equiv \int_0^L u_{\text{elastic}}(y)\, dy \tag{49b}$$

Hence, it is possible to compute an effective elastic constant k by

$$k_{\text{Dumbbell}} \equiv \frac{\text{Force}}{\delta} = -\frac{EI_{j=3} X^{(iii)}_{j=3}(L)}{\delta} \tag{50}$$

Replacing k_{pend} with k_{Dumbbell} in the matrix $\mathcal{K}$ of Eq.45 and then computing the impedance matrix Z as in Eq.44, it is possible to evaluate the thermal noise in Eq.43. The code that implements the optimised fibre model is in Appendix M and the structure that contains the geometrical description of the dumbbell shaped fibre is **ifo.Suspension.Stage(1)**.

3.2 Vertical bouncing mode thermal noise

The computation of the vertical thermal noise of the coupled oscillators system (Marionette–Reference Mass–Mirror) is performed in Sec.(1.5) of [26]. Essentially is applied again Eq.43 where the dissipation matrix Γ_v and the frequencies matrix Ω_v are

$$\Gamma_v = \begin{pmatrix} \gamma_{v\,1} & 0 & -\gamma_{v\,1} \\ 0 & \gamma_{v\,2} & -\gamma_{v\,2} \\ -\frac{\mu_1}{\mu_3}\gamma_{v\,1} & \frac{\mu_2}{\mu_3}\gamma_{v\,2} & \frac{\mu_3\gamma_{v\,3} - \mu_1\gamma_{v\,1} - \mu_2\gamma_{v\,2}}{\mu_3} \end{pmatrix} \tag{51a}$$

$$\Omega_v = \begin{pmatrix} \omega^2_{v\,1} & 0 & -\omega^2_{v\,1} \\ 0 & \omega^2_{v\,2} & -\omega^2_{v\,2} \\ -\frac{\mu_1}{\mu_3}\omega^2_{v\,1} & -\frac{\mu_2}{\mu_3}\omega^2_{v\,2} & \frac{\mu_1\omega^2_{v\,1} + \mu_2\omega^2_{v\,2} + \omega^2_{v\,3}}{\mu_3} \end{pmatrix} \tag{51b}$$

where

$$\omega^2_{v\,j} = \begin{cases} \frac{E_j(\pi r_j^2)}{L_j} & \text{if } j = 1, 2 \\ (2\pi f_{v\,SA})^2 & \text{if } j = 3 \end{cases} \tag{52}$$

Here $f_{v\,SA}$ is the vertical mode of the whole Super–Attenuator (SA) (in Virgo $f_{v\,SA} \simeq 4\,\text{Hz}$) dominated by the softness of the anti–spring system.

4 Residual Gas phase noise computation

The evaluation of the residual gas phase noise is based on the computation made by Whitcomb and Weiss [30, 31], verified by Zucker [32] and finally adapted for a multi–gas residual contamination by Cella [33]. In the hypothesis of absence of collision between the gas particles and the boundaries, absence of delays and interaction with the beam, the power spectrum of the equivalent length fluctuation is (considering 2 arms):

$$S_L(f) = \sum_A \frac{8\rho_A(2\pi\alpha_A)^2}{V_{p,A}} \int_0^L \frac{1}{w(z)} \exp\left[-2\pi f \frac{w(z)}{V_{p,A}}\right] dz \tag{53}$$

where $V_{p,A}$ is the most probable velocity of the gas molecula of mass M_A:

$$V_{p,A} = \sqrt{\frac{2k_BT}{M_A}} \tag{54}$$

ρ_A is the number density of the gas, related to the partial pressure P_A of the gas by:

$$\rho_A = \frac{P_A}{k_BT} \tag{55}$$

The summ of formula 53 is extended to all the gasses playing a role in determining the residual pressure. In AdV are taken in account H_2O, H_2, N_2, O_2 and the properties of thes gasses are included in the Matlab data structure named **ifo.Infrastructure.ResidualGas** and listed in Appendix A. The code implementing the Residual Gas computation is shown in the Appendix N.

5 Quantum Noise

In the initial interferometers, like Virgo, the description of the noise related to the discrete photon composition of the light has been treated in a classical way, by distinguishing the shot noise, due to the photon counting, from the radiation pressure noise, due the photon impulse transfer to the mirror. This naive description of the quantum noise fails in the case of the advanced detectors, because of the presence of the signal recycling mirror (SRM) at the output port. The effect of the SRM is to correlate the shot noise to the radiation pressure through the mixing of the quadratures at the output of the interferometer; in this case it is better to use the appropriate name of quantum noise.

The description of the quantum noise in an advanced interferometer is performed in two fundamental papers: H. J. Kimble and collaborators [34] (usually indicated as KLMTV), where the right formalism is introduced, and A. Buonanno and Y.Chen [35], where the full computation is performed. Here we will report only the results, using the same formalism and, as much as possible, symbols adopted in these articles.

5.1 Initial interferometers

Let start from a non–signal recycled interferometer (like Virgo). The spectral amplitude of optical read-out noise, using the KLMTV formalism, can be written as:

$$h_{or}(\omega) = \frac{h_{SQL}(\omega)}{\sqrt{2}} \sqrt{\frac{1}{\mathcal{K}(\omega)} + \mathcal{K}(\omega)} \tag{56}$$

where

$$h_{SQL}(\omega) = \sqrt{\frac{8\hbar}{m\omega^2L^2}} \tag{57}$$

is the Standard Quantum Limit (SQL) (m here is the mirror mass),

$$\mathcal{K}(\omega) = \frac{I_{BS}}{I_{SQL}} \frac{2\gamma_{FP}^4}{\omega^2(\gamma_{FP}^2 + \omega^2)} \tag{58}$$

is the effective coupling constant between the motion of the test mass and the output signal, having defined I_{BS} the input power impinging on the beam splitter (BS) mirror,

$$\gamma_{FP} = 2\pi f_{FP} = 2\pi \frac{c}{4L \cdot \mathcal{F}} = \frac{T \cdot c}{4L} \tag{59}$$

is the decay factor of the Fabry–Perot (FP) cavity, where $\mathcal{F}$ is the finesse of the main cavities and T is the power transmittivity of the input mirrors. I_{SQL} is the light power needed to reach the SQL at a (GW) frequency $\omega = \gamma_{FP}$:

$$I_{SQL} = \frac{mL^2\gamma_{FP}^4}{4\omega_{\rm laser}} \;\; ; \;\; \omega_{\rm laser} = \frac{2\pi c}{\lambda} \tag{60}$$

By noting that $\mathcal{K} \propto m^{-1}$ and $h_{SQL} \propto m$ it is easy to see that in Eq.56 the first term within the root gives the shot noise contribution (independent from m) and the second term gives the radiation pressure ($\propto 1/m\omega^2$).

5.2 Read–out schemes

In effect the read–out scheme of initial GW detectors was based on a Heterodyne configuration (see left panel of Fig.7) and the shot noise part should be increased by the non-stationary contribution given by the radio–frequency (RF) modulation (computed to be $\sqrt{3/2}$ in Virgo [36, 37]). In theory, the SQL is an effective limit for an interferometer with free floating mass, but thanks to the ponderomotive squeezing effect due to the fluctuating radiation pressure the SQL can be overcome, because of the correlation introduced between the radiation pressure and the phase (shot noise) components. This needs a different read-out scheme, the homodyne detection, shown in the central panel of Fig.7, being it one of the quantum nondemolition devices [38]. Using an homodyne detection scheme the optical read–out noise can be written [35] as

$$h_{or}(\omega) = \frac{h_{SQL}(\omega)}{\sqrt{2\mathcal{K}}} \sqrt{1 + (\tan\eta - \mathcal{K})^2} \tag{61}$$

where η is the homodyne phase; it is evident that for $\eta = \arctan(\mathcal{K})$ the SQL is overcome.

Because of the technical difficulties [39] to implement a complete homodyne read–out scheme in GW detectors, a special case of homodyne is implemented: the DC detection (see right panel of Fig.7). In a DC detection scheme the operating point of the interferometer is slightly shifted off the resonance by acting either on the MICH or on the DARM control loops [40]. The small amount of carrier light exiting at the output port because of this offset acts as local oscillator beating with GW-produced field from differential motion of the arms at GW (audio) frequencies. The dark–fringe offset acts as homodyne phase (disturbed by all the effects of unbalancing between the cavities).

5.3 Signal Recycled interferometer

Let add a mirror at the exit port of the interferometer, at a distance l from the Beam Splitter mirror, as shown in Fig.8. This mirror reinject the electric fields d_i (the index i indicates the two quadratures) back in to the interferometer. Since one of these two quadrature contains the gravitational signal, this configuration is generically named "Signal Recycled" (SR). The full description of the of a SR interferometer is performed in two twin articles [35, 41] and here we report the results and few simplified considerations.

Let consider $i = 1$ the amplitude quadrature and $i = 2$ the phase quadrature. In a conventional interferometer the antisimmetric gravitational signal is embedded in d_2 (see Fig.8). The effect of the SR mirror is to reflect back the fields d_1 and d_2 mixing up their content [42]:

$$\begin{pmatrix} \widehat{d}_1 \\ \widehat{d}_2 \end{pmatrix} \xrightarrow{\text{SR cavity}} \rho \begin{pmatrix} \cos 2\phi & -\sin 2\phi \\ \sin 2\phi & \cos 2\phi \end{pmatrix} \begin{pmatrix} \widehat{d}_1 \\ \widehat{d}_2 \end{pmatrix} + \tau \begin{pmatrix} \text{vacuum fields} \\ \text{from outside} \end{pmatrix} \Leftrightarrow \begin{pmatrix} \widehat{c}_1 \\ \widehat{c}_2 \end{pmatrix} \tag{62}$$

where ρ and τ here are the amplitude reflectivity and transmissivity of the SR mirror and

$$\phi \equiv \left[\frac{\omega_{\rm laser} l}{c}\right]_{\rm mod 2\pi} = \left[2\pi \frac{l}{\lambda}\right]_{\rm mod 2\pi} \tag{63}$$

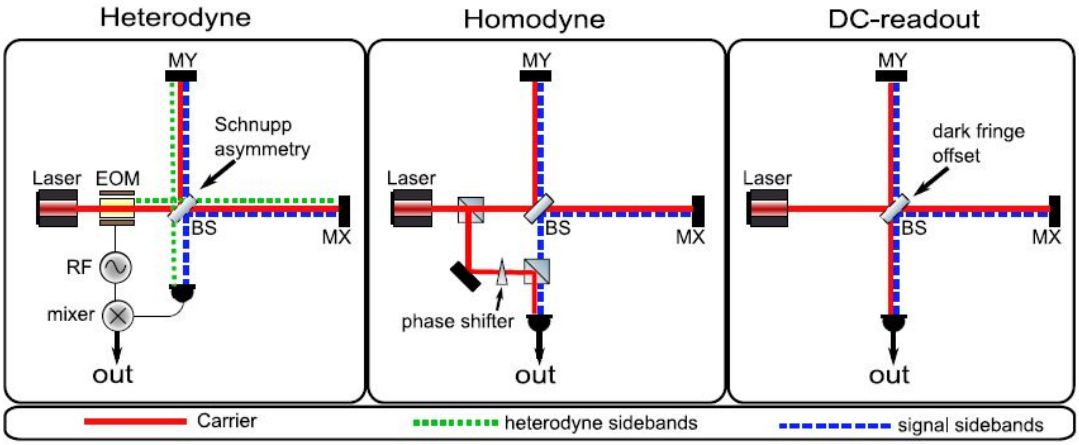

Figure 7: Figure extracted from [39]. Read–Out schemes for GW interferometers; from left to right: Heterodyne, Homodyne and DC detection.

is the phase gained by the carrier frequency $\omega_{\rm laser}$ while travelling one way in the SR cavity. The corresponding phase gained by the GW sidebands is $\Phi \equiv [\omega l/c]_{\rm mod2\pi}$. Eq.62 suggests that, with in the SR interferometer, both the output quadratures b_i contain the antisimmetric h signal and it is impossible to put the signal info in only one by a transformation. The role of the homodyne detection, then, is to extract a linear combination of the two quadratures:

$$b_\eta = b_1 \sin\eta + b_2 \cos\eta \tag{64}$$

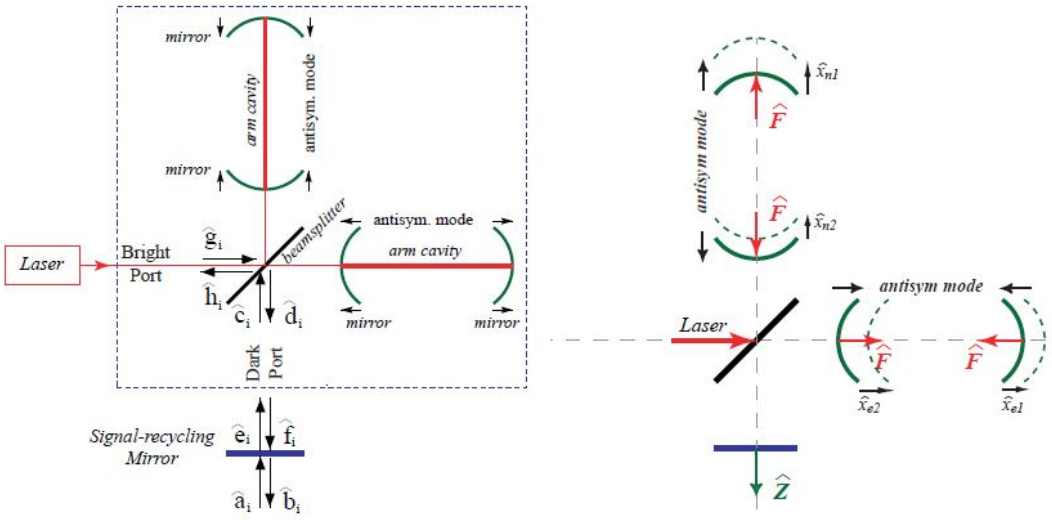

Figure 8: Figure extracted from [42]. In the left panel is schematised a signal recycles (SR) interferometer with all the electric fields $(a_i, b_i, \dots\ , \ i = 1, 2)$. In the right panel the variables needed to define the antisymmetric mode (containing the gravitational signal) are shown: $\hat{x}_{\rm antisym} = (\hat{x}_{n1} - \hat{x}_{n2}) - (\hat{x}_{e1} - \hat{x}_{e2})$.

Now we are almost ready to jump to the conclusions of the Buonanno and Chen paper [35], but first we need to introduce the losses in the interferometer (as done in KLMTV [34] and extended in [35]). The losses in the SR cavities are introduced as an extra field p_i entering from the SR mirror and the losses in the output

photo–detection process are introduced as an addition output field q_i (see Fig.7 of [35]). As done in KLMTV, at the losses in arm cavities correspond a vacuum field (having quadratures n_i) entering in the interferometer; these losses are quantified by the loss coefficient per round trip (RTL) $\mathcal{L}$. The parameter ϵ

$$\epsilon \equiv \frac{2\mathcal{L}}{T} = \frac{\mathcal{L}}{2\gamma_{FP}L/c} \tag{65}$$

represents the fraction of carrier photons, impinging in each cavity, that gets lost. It is useful also to define the frequency dependent parameter

$$\mathcal{E} = \frac{2\epsilon}{1+(\omega_{\rm laser}/\gamma_{FP})^2} \tag{66}$$

The photodetector loss is defined by λ_{PD} and the losses inside the SR cavity by the fraction of photons lost at each bounce of the interior field off the SR mirror λ_{SR}. The input–output relations for the lossy field b_i^L at the SRM are given by Eq. 5.6 in [35], hereafter copied:

$$\begin{aligned}\begin{pmatrix} b_1^L \\ b_2^L \end{pmatrix} = \frac{1}{M^L}\Bigg[& e^{2i\beta}\begin{pmatrix} C_{11}^L & C_{12}^L \\ C_{21}^L & C_{22}^L \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \sqrt{2\mathcal{K}}\tau e^{i\beta}\begin{pmatrix} D_1^L \\ D_2^L \end{pmatrix}\frac{h}{h_{SQL}} + \\ & e2i\beta\begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + e^{2i\beta}\begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + e^{2i\beta}\begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix}\begin{pmatrix} n_1 \\ n_2 \end{pmatrix}\Bigg]\end{aligned} \tag{67}$$

The quantum noise power spectrum is then given by the Eq.5.13 of [35], hereafter copied:

$$\begin{aligned} S_h^\eta = \frac{h_{SQL}^2}{2\mathcal{K}\tau^2|D_1^L\sin\eta + D_2^L\cos\eta|^2}\Big[& |C_{11}^L\sin\eta + C_{21}^L\cos\eta|^2 + |C_{12}^L\sin\eta + C_{22}^L\cos\eta|^2 + \\ & |P_{11}\sin\eta + P_{21}\cos\eta|^2 + |P_{12}\sin\eta + P_{22}\cos\eta|^2 + \\ & |Q_{11}\sin\eta + Q_{21}\cos\eta|^2 + |Q_{12}\sin\eta + Q_{22}\cos\eta|^2 + \\ & |N_{11}\sin\eta + N_{21}\cos\eta|^2 + |N_{12}\sin\eta + N_{22}\cos\eta|^2\Big]\end{aligned} \tag{68}$$

The matrix elements used in Eq.67 and Eq.68 are defined in the Eq.5.7–5.12 of [35], hereafter copied:

$$\begin{aligned} M^L = 1 + \rho^2 e^{4i\beta} - 2\rho\left(\cos 2\phi + \frac{\mathcal{K}}{2}\sin 2\phi\right)e^{2i\beta} + \lambda_{SR}\,\rho\left(-\rho e^{2i\beta} + \cos 2\phi + \frac{\mathcal{K}}{2}\sin 2\phi\right)e^{2i\beta} \\ + \epsilon\rho\left[2\left(\cos\beta\right)^2\left(-\rho e^{2i\beta} + \cos 2\phi\right)\frac{\mathcal{K}}{2}\left(3 + e^{2i\beta}\right)\sin 2\phi\right]e^{2i\beta}\end{aligned} \tag{69a}$$

$$\begin{aligned} C_{11}^L = C_{22}^L = \sqrt{1-\lambda_{PD}}\Bigg\{ & \left(1+\rho^2\right)\left(\cos 2\phi + \frac{\mathcal{K}}{2}\sin 2\phi\right) - 2\rho\cos 2\beta \\ & -\frac{1}{4}\epsilon\left[-2\left(1+e^{2i\beta}\right)^2\rho + 4\left(1+\rho^2\right)\cos^2\beta\cos 2\phi + \left(3+e^{2i\beta}\right)\mathcal{K}\left(1+\rho^2\right)\sin 2\phi\right] \\ & +\lambda_{SR}\left[e^{2i\beta}\rho - \frac{1}{2}\left(1+\rho^2\right)\left(\cos 2\phi + \frac{\mathcal{K}}{2}\sin 2\phi\right)\right]\Bigg\}\end{aligned} \tag{69b}$$

$$\begin{aligned} C_{12}^L = \sqrt{1-\lambda_{PD}}\,\tau^2\Big\{ & -\left(\sin 2\phi + \mathcal{K}\sin^2\phi\right) \\ & +\frac{1}{2}\epsilon\sin\phi\left[\left(3+e^{2i\beta}\right)\mathcal{K}\sin\phi + 4\cos^2\beta\cos\phi\right] + \frac{1}{2}\lambda_{SR}\left(\sin 2\phi + \mathcal{K}\sin^2\phi\right)\Big\}\end{aligned} \tag{69c}$$

$$C_{21}^{L} = \sqrt{1-\lambda_{PD}}\,\tau^2\left\{\left(\sin 2\phi - \mathcal{K}\cos^2\phi\right)\right.$$
$$\left.+\frac{1}{2}\epsilon\cos\phi\left[\left(3+e^{2i\beta}\right)\mathcal{K}\cos\phi - 4\cos^2\beta\sin\phi\right] + \frac{1}{2}\lambda_{SR}\left(-\sin 2\phi + \mathcal{K}\cos^2\phi\right)\right\} \quad (69d)$$

$$D_1^{L} = \sqrt{1-\lambda_{PD}}\left\{-\left(1+\rho e^{2i\beta}\right)\sin\phi + \frac{1}{4}\epsilon\left[3+\rho+2\rho e^{4i\beta}+e^{2i\beta}\left(1+5\rho\right)\right]\sin\phi + \frac{1}{2}\lambda_{SR}\,e^{2i\beta}\rho\sin\phi\right\} \quad (69e)$$

$$D_2^{L} = \sqrt{1-\lambda_{PD}}\left\{-\left(-1+\rho e^{2i\beta}\right)\cos\phi + \frac{1}{4}\epsilon\left[-3+\rho+2\rho e^{4i\beta}+e^{2i\beta}\left(-1+5\rho\right)\right]\cos\phi + \frac{1}{2}\lambda_{SR}\,e^{2i\beta}\rho\cos\phi\right\} \quad (69f)$$

$$P_{11} = P_{22} = \frac{1}{2}\sqrt{1-\lambda_{PD}}\,\sqrt{\lambda_{SR}}\,\tau\left(-2\rho e^{2i\beta} + 2\cos 2\phi + \mathcal{K}\sin 2\phi\right) \quad (69g)$$

$$P_{12} = -\sqrt{1-\lambda_{PD}}\,\sqrt{\lambda_{SR}}\,\tau\sin\phi\left(2\cos\phi + \mathcal{K}\sin\phi\right) \quad (69h)$$

$$P_{21} = \sqrt{1-\lambda_{PD}}\,\sqrt{\lambda_{SR}}\,\tau\cos\phi\left(2\sin\phi - \mathcal{K}\cos\phi\right) \quad (69i)$$

$$Q_{11} = Q_{22} = \sqrt{\lambda_{PD}}\left\{e^{-2i\beta} + \rho^2 e^{2i\beta} - \rho\left(2\cos 2\phi + \mathcal{K}\sin 2\phi\right)\right.$$
$$+\frac{1}{2}\epsilon\rho\left[e^{-2i\beta}\cos 2\phi + e^{2i\beta}\left(-2\rho - 2\rho\cos 2\beta + \cos 2\phi + \mathcal{K}\sin 2\phi\right) + 2\cos 2\phi + 3\mathcal{K}\sin 2\phi\right]$$
$$\left.-\frac{1}{2}\lambda_{SR}\,\rho\left[2\rho e^{2i\beta} - 2\cos 2\phi - \mathcal{K}\sin 2\phi\right]\right\} \quad (69j)$$

$$Q_{12} = 0 = Q_{21} \quad (69k)$$

$$N_{11} = \sqrt{1-\lambda_{PD}}\,\sqrt{\frac{\epsilon}{2}}\,\tau\,\left\{\mathcal{K}\left(1+\rho e^{2i\beta}\right)\sin\phi + 2\cos\beta\left[e^{-i\beta}\cos\phi - \rho e^{i\beta}\left(\cos\phi + \mathcal{K}\sin\phi\right)\right]\right\} \quad (69l)$$

$$N_{22} = -\sqrt{1-\lambda_{PD}}\sqrt{2\epsilon}\,\tau\,\left(-e^{-i\beta} + \rho e^{i\beta}\right)\cos\beta\,\cos\phi \quad (69m)$$

$$N_{12} = -\sqrt{1-\lambda_{PD}}\sqrt{2\epsilon}\,\tau\,\left(e^{-i\beta} + \rho e^{i\beta}\right)\cos\beta\,\sin\phi \quad (69n)$$

$$N_{21} = \sqrt{1-\lambda_{PD}}\,\sqrt{\frac{\epsilon}{2}}\,\tau\,\left\{-\mathcal{K}\left(1+\rho\right)\cos\phi + 2\cos\beta\left(e^{-i\beta} + \rho e^{i\beta}\right)\cos\beta\,\sin\phi\right\} \quad (69o)$$

In the previous equations β indicates the phase shift of the GW sideband in the arm:

$$\beta = \arctan\frac{\omega}{\gamma_{FP}} \quad (70)$$

The computation of the quantum noise in the gwinc code is shown in Appendix O.

5.4 Optical losses budget

To compute the quantum noise it is crucial to evaluate the RTL losses $\mathcal{L}$, the photo–detection process losses λ_{PD} and the SRC losses λ_{SR}. In AdV the RTL losses, requirement is

$$\mathcal{L} \leq 75 \text{ ppm} \quad (71)$$

(50 ppm linked to flatness defects, 25 ppm due to transmission, absorption, scattering); this is coded in the variable **ifo.Optics.Loss** in the Appendix A that contains the single mirror losses (= $\mathcal{L}/2$).In effect in gwinc, to these losses is added contribution of the clipping losses (see Appendix P):

$$\mathcal{L}_{\text{clip}} = \exp\left[-2\left(\frac{R_{\text{coat}}^{\text{ITM}}}{R_{\text{beam}}^{\text{ITM}}}\right)^2\right] + \exp\left[-2\left(\frac{R_{\text{coat}}^{\text{ETM}}}{R_{\text{beam}}^{\text{ETM}}}\right)^2\right] \quad (72)$$

Table 2: Computation of the Photo-detection process loss budget (λ_{PD})

Process	**Comment**	**losses**
Telescope aberrations		$\leq 0.5\%$
OMC losses	mode mismatch, misalignment, surface losses	$\leq 5.0\%$
Faraday Isolator		$= 2.0\%$
Sub–total optical losses		$\leq 7.5\%$
B1p photodiodes	$2 \times 2.5\,\mathrm{mW}$ (only $0.24\,\mathrm{mW}$ on TEM00)	$= 0.3\%$
B1p quadrants	$2 \times 2.5\,\mathrm{mW}$ (only TEM00 + SB) $\Rightarrow$ $1.6\,\mathrm{mW}$ on TEM00	$= 2.0\%$
B1 photodiodes with AC signals	$0.8\,\mathrm{mW}$?	$\simeq 1.0\%$?
B1p phase camera	$15\,\mathrm{mW}$ (only $0.8\,\mathrm{mW}$ on B1?)	$\simeq 1.0\%$?
Sub–total sensing losses		$\leq 4.5\%$
B1 DC Photodiodes	QE ≥ 0.98 in GEO	$\leq 2.0\%$
Total PD losses λ_{PD}		$\leq 14\%$

The computation of the λ_{PD} losses are based on [43], reproduced in Table 5.4 and coded in the variable **ifo.Optics.PhotoDetectorEfficiency** in Appendix A.

The SRC losses λ_{SR} are computed in Appendix O as the pile–up of 3 three contributions: the mismatch between the modes in the interferometer arms and in the SRC **ifo.Optics.coupling** , set to be $1-0.997 = 0.3\,\%$; the losses around the BS mirror **ifo.Optics.BSLoss** , set to be 1510ppm (see Table 5.4); the losses due to the residual thermal lensing due to the limits of the thermal compensation system (TCS) **ifo.TCS.SRCloss**, currently set to 300ppm (see TCS chapter of [44]).

Table 3: Computation of the PRC loss budget (table extracted from the Optical Simulation and Design chapter of [44])

Loss origin	**Loss per round trip** [ppm]
Pick–Off Plate HR	2×300
Pick–Off Plate AR	2×300
2 Compensation Plates	2×200
Beam Splitter AR	100
Surface scattering	17×10
Absorption	40
Losses in PRC	1510

6 AdV Sensitivity

The final AdV sensitivity, computed through gwinc, is shown in Fig.9. The expected performances, in this provisional configuration, are:

- NS–NS detection distance: $D_{NS} \simeq 134\,\mathrm{Mpc}$
- BH–BH detection distance: $D_{BH} \simeq 1015\,\mathrm{Mpc}$

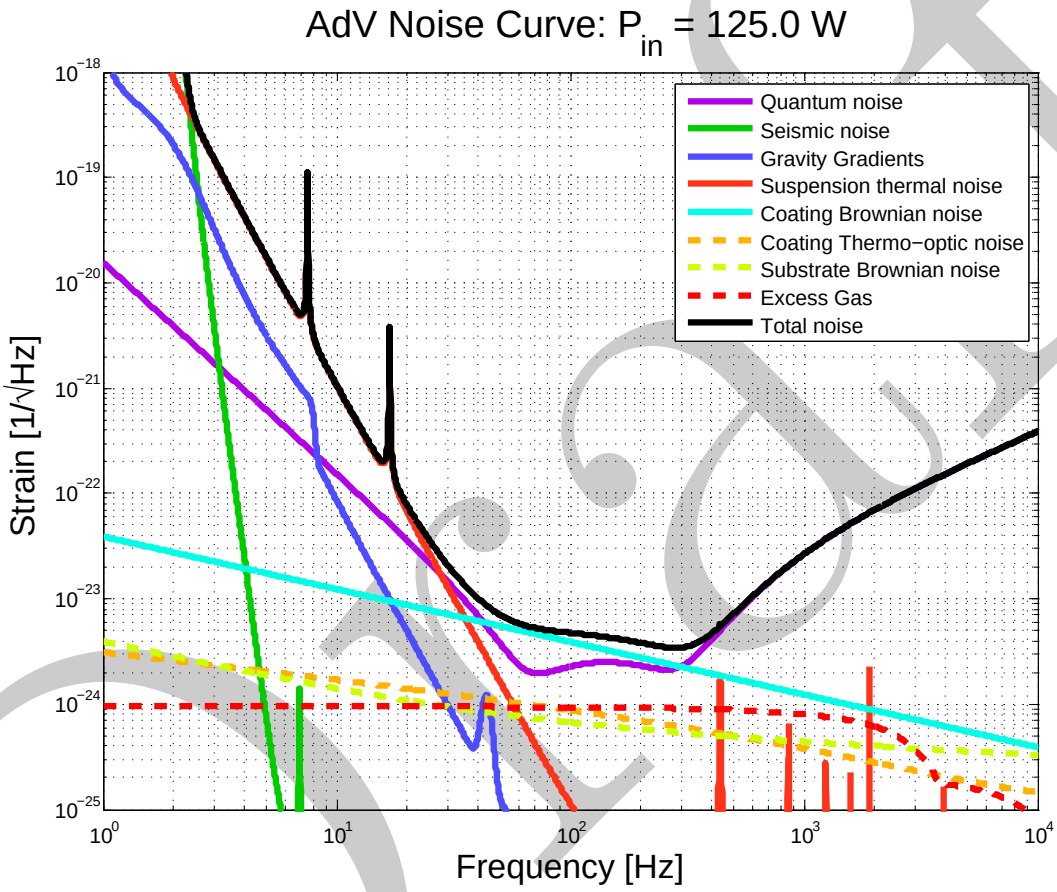

Figure 9: AdV sensitivity computed by gwinc with the parameters in Appendix A.

References

[1] P. R. Saulson, "Terrestrial gravitational noise on a gravitational wave antenna," *Phys. Rev. D*, vol. 30 (4), pp. 732–736, 1984. 5, 6

[2] M. Beccaria et al., "Relevance of Newtonian seismic noise for the VIRGO interferometer sensitivity," *Classical and Quantum Gravity*, vol. 15, pp. 3339–3362, 1998. 6

[3] S. A. Hughes and K. S. Thorne, "Seismic gravity-gradient noise in interferometric gravitational-wave detectors," *Phys. Rev. D*, vol. 58 (12), p. 122002, 1998. 5, 6

[4] M. W. Coughlin and I. Fiori, "Characterization study of the Virgo seismic environment," Tech. rep., Virgo internal note VIR-0489A-10, 2010. 5

[5] P. Ruggi, "Low frequency seismic noise: locking and sensitivity issues," Tech. rep., Virgo internal note VIR-0882A-07, 2007. 5

[6] P. Ruggi, "private communication," 2012. 5, 6, 9, 10

[7] I. Fiori, "Private communication," 2012. 5, 6, 7, 8

[8] F. Acernese et al., "Measurements of Superattenuator seismic isolation by Virgo interferometer," *Astroparticle Physics*, vol. 33 (3), pp. 182 – 189, 2010. 5

[9] P. Ruggi, *L'attenuazione del rumore sismico nel rivelatore di onde gravitazionali Virgo*, Ph.D. thesis, University of Pisa, 2003. 6

[10] T. E. science team, "Einstein gravitational wave Telescope conceptual design study," 2011, eT internal note: ET-0106C-10. 6

[11] D. Hillel, *Fundamentals of soil physics*, Physiological ecology, Academic Press, 1980. 6

[12] "Proprietá fisiche del terreno," Wikipedia, 2012. 6

[13] H. B. Callen and R. F. Greene, "On a Theorem of Irreversible Thermodynamics," *Phys. Rev.*, vol. 86, pp. 702–710, 1952. 11, 14

[14] Y. Levin, "Internal thermal noise in the LIGO test masses: A direct approach," *Phys. Rev. D*, vol. 57, pp. 659–663, 1998. 11

[15] F. Bondu, P. Hello, and J.-Y. Vinet, "Thermal noise in mirrors of interferometric gravitational wave antennas," *Physics Letters A*, vol. 246, pp. 227 – 236, 1998. 11

[16] J.-Y. Vinet, "Noises produced by opto-thermal couplings in Mirrors," 2001, vIR-NOT-OCA-1390-166. 11

[17] N. Nakagawa, A. M. Gretarsson, E. K. Gustafson, and M. M. Fejer, "Thermal noise in half-infinite mirrors with nonuniform loss: A slab of excess loss in a half-infinite mirror," *Phys. Rev. D*, vol. 65, p. 102001, 2002. 12, 13

[18] S. D. Penn et al., "Frequency and surface dependence of the mechanical loss in fused silica," *Physics Letters A*, vol. 352 (1âĂŞ2), pp. 3 – 6, 2006. 12

[19] Y. T. Liu and K. S. Thorne, "Thermoelastic noise and homogeneous thermal noise in finite sized gravitational-wave test masses," *Phys. Rev. D*, vol. 62, p. 122002, 2000. 13

[20] V. Braginsky, M. Gorodetsky, and S. Vyatchanin, "Thermodynamical fluctuations and photo-thermal shot noise in gravitational wave antennae," *Physics Letters A*, vol. 264 (1), pp. 1 – 10, 1999. 14

[21] V. Braginsky and S. Vyatchanin, "Thermodynamical fluctuations in optical mirror coatings," *Physics Letters A*, vol. 312 (3âĂŞ4), pp. 244 – 255, 2003. 16

[22] M. M. Fejer et al., "Thermoelastic dissipation in inhomogeneous media: loss measurements and displacement noise in coated test masses for interferometric gravitational wave detectors," *Phys. Rev. D*, vol. 70, p. 082003, 2004. 14

[23] J. Franc et al., "Mirror thermal noise in laser interferometer gravitational wave detectors operating at room and cryogenic temperature," Tech. rep., Einstein Telescope internal note ET-021-09, 2009, eT-021-09. 14

[24] V. Braginsky, M. Gorodetsky, and S. Vyatchanin, "Thermo-refractive noise in gravitational wave antennae," *Physics Letters A*, vol. 271 (5âĂŞ6), pp. 303 – 307, 2000. 14

[25] M. Evans et al., "Thermo-optic noise in coated mirrors for high-precision optical measurements," *Phys. Rev. D*, vol. 78, p. 102003, 2008. 14

[26] F. Piergiovanni, M. Punturo, and P. Puppo, "The thermal noise of the Virgo+ and Virgo Advanced Last Stage Suspension (The PPP effect)." Tech. rep., Virgo internal note VIR-0015E-09, 2009. 17, 18, 19, 36

[27] G. Cagnoli and P. A. Willems, "Effects of nonlinear thermoelastic damping in highly stressed fibers," *Phys. Rev. B*, vol. 65, p. 174111, 2002. 18

[28] G. I. González and P. R. Saulson, "Brownian motion of a mass suspended by an anelastic wire," *The Journal of the Acoustical Society of America*, vol. 96 (1), pp. 207–212, 1994. 18

[29] P. Willems, "Dumbbell-shaped fibers for gravitational wave detectors," *Physics Letters A*, vol. 300 (2âĂŞ3), pp. 162 – 168, 2002. 19

[30] S. E. Whitcomb, "Optical pathlength fluctuations in an interferometer due to residual gas," Tech. rep., California Institute of Technology, 1984. 20

[31] R. Weiss, "Scattering by residual gas," Tech. rep., Massachusetts Institute of Technology, 1989. 20

[32] M. E. Zucker and S. E. Whitcomb, "Measurement of Optical Path Fluctuations due to Residual Gas in the LIGO 40 Meter Interferometer," Tech. Rep. P940008, LIGO internal note, 1991. 20

[33] G. Cella, "Residual Pressure Effects," 2008, virgo Bi-weekly meeting presentation. 20

[34] H. J. Kimble, Y. Levin, A. B. Matsko, K. S. Thorne, and S. P. Vyatchanin, "Conversion of conventional gravitational-wave interferometers into quantum nondemolition interferometers by modifying their input and/or output optics," *Phys. Rev. D*, vol. 65, p. 022002, 2001. 20, 22

[35] A. Buonanno and Y. Chen, "Quantum noise in second generation, signal-recycled laser interferometric gravitational-wave detectors," *Phys. Rev. D*, vol. 64, p. 042006, 2001. 20, 21, 22, 23

[36] F. Bondu, "Dark Fringe Shot Noise Sensitivity," Tech. rep., Virgo internal note VIR-NOT-OCA-1390-243, 2003. 21

[37] M. Punturo, "The Virgo sensitivity curve," Tech. rep., Virgo internal note VIR-NOT-PER-1390-51, 2004. 21

[38] V. Braginsky, *Sov. Phys.—JETP*, vol. 26, p. 831, 1968. 21

[39] S. Hild et al., "DC-readout of a signal-recycled gravitational wave detector," *Classical and Quantum Gravity*, vol. 26 (5), p. 055012, 2009. 21, 22

[40] B. Abbott et al., "Proposal for a Homodyne (DC) Detection Experiment at the LIGO Caltech 40-Meter Laboratory," Tech. rep., LIGO internal note LIGO-T050086-00-R, 2005. 21

[41] A. Buonanno and Y. Chen, "Signal recycled laser-interferometer gravitational-wave detectors as optical springs," *Phys. Rev. D*, vol. 65, p. 042001, 2002. 21

[42] A. Buonanno and Y. Chen, "Signal recycled laser-interferometer gravitational-wave detectors as optical springs," *arXiv:gr-qc/0107021v1*, 2001. 21, 22

[43] R. Gouaty, "private communication," 2012. 25

[44] The Virgo Collaboration, "Advanced Virgo Technical Design Report," Tech. rep., Virgo internal note VIR-xxxA-12, 2012. 25, 39

[45] D. R. M. Crooks et al., "Experimental measurements of mechanical dissipation associated with dielectric coatings formed using SiO 2 , Ta 2 O 5 and Al 2 O 3," *Classical and Quantum Gravity*, vol. 23 (15), p. 4953, 2006. 38

Nomenclature

Abbreviations

AdV Advanced Virgo

aLIGO Advanced LIGO

BS Beam Splitter

CB Virgo Central Building

FP Fabry–Perot

IP Inverted Pendulum

RTL Round Trip Losses

SA Super–Attenuator, the passive filtering system of Virgo

SQL Standard Quantum Limit

SRM Signal Recycling Mirror

TF Transfer Function

V+ Virgo+

Glossary

Compact binary A compact binary is an astronomical binary consisting of a pair of compact stars

Compact star Throughout this document a compact star stands for a neutron star or a black hole

Dual Recycling Using Power- and Signal-Recycling at the same time

G1 The GEO600 detector near Hannover in Germany

GEO600 The GEO600 detector located near Hannover in Germany

H1 The LIGO 4-kilometre interferometer at Hanford, USA

H2 The LIGO 2-kilometre interferometer at Hanford, USA

Horizon Distance The distance at which a gravitational wave detector would measure a matched-filter signal-to-noise ratio of 8 for an optimally oriented (i.e. face-on) compact binary source located in a direction perpendicular to the plane of the detector

L1 The LIGO 4-kilometre interferometer at Livingston Parish, Louisiana, USA

Power Recycling Re-using the light reflected back to the interferometer input by placing a mirror there and resonantly enhancing the circulating light power. Has the same effect as using a more powerful laser.

PPP model Thermal noise model describing the contribution of the whole last stage of the suspension (including the Marionette, the Reference Mass and the Mirror), developed by F. Piergiovanni, M. Punturo and P. Puppo

QND Typically the term quantum nondemolition (QND) measurement is used to describe a measurement of a quantum system which preserves the integrity of the system and the value of the measured observable. In the literature on gravitational wave detectors this term is often used to describe a variety of interferometer schemes in which shot noise and radiation pressure noise can be simultaneously suppressed. Such systems are typically not performing a strict QND measurement, thus they may more appropriately be referred to as Quantum Noise Reduction (QNR) systems.

Resonant Sideband Extraction The same technique as SR but operated anti-resonant, i.e. widening and/or detuning the bandwidth by reducing the reflectivity of the compound mirror formed by the inboard cavity mirror and the SR mirror.

Signal Recycling Resonantly enhancing the GW signals exiting the interferometer through the output port by placing a mirror there. This increases the sensitivity at the cost of the bandwidth. With a different (anti-resonant) tuning the same technique can be used for widening the bandwidth at the expense of the sensitivity (RSE). SR can optimize the sensitivity for an arbitrary frequency.

V1 The Virgo gravitational wave detector in Italy

Vacuum fluctuations Fluctuations that result from the quantum nature of an electromagnetic field even at the lowest possible energy level (zero mean energy = vacuum).

Virgo Virgo is a 3-kilometre gravitational wave detector located near Pisa, Italy

Symbols

Please note that some symbols might stand for more than one quantity depending on the context

α_A Polarizability of the residual gas molecula

β dn/dT refraction index thermal gradiend

C Heat capacity per volume unit $\left[\mathrm{JK^{-1}m^{-3}}\right]$

D_{L} Luminosity distance to the source

E_{FS} Young modulus of the Fused Silica substrate

$\mathcal{F}$ Finesse of a Fabry Perot cavity

h Gravitational wave amplitude, usually denoting the detector response

κ Thermal conductivity $\left[\mathrm{WK^{-1}m^{-1}}\right]$

$\mathcal{K}$ Effective coupling constant between the motion of the test mass and the output signal

k_B Boltzman Constant $k_B = 1.380658 \times 10^{-23}\,\mathrm{J/K}$

L Geometric length

λ Laser wavelenght $\lambda = 1064\mathrm{nm}$

σ_{FS} Poisson ration of the Fused Silica substrate

$w(z)$ beam radius profile

Appendices

A IFO structure

```
function ifo =IFOModel(varargin)

% IFOMODEL returns a structure describing an IFO for use in
% benchmark programs and noise simulator. Part of the bench
% package, which provides science-grounded figures of merit for
% comparing interferometric gravitational wave detector designs.
%

% Id: IFOModel.m,C_v00r03 2012/07/24
% VIR-0073C-12
% * References:
% * 1. Electro-Optic Handbook, Waynant & Ediger (McGraw-Hill: 1993)
% * 2. LIGO/GEO data/experience
% * 3. Suspension reference design, LIGO-T000012-00
% * 4. Quartz Glass for Optics Data and Properties, Heraeus data sheet,
% *    numbers for suprasil
% * 5. Y.S. Touloukian (ed), Thermophysical Properties of Matter
% *    (IFI/Plenum,1970)
% * 6. Marvin J. Weber (ed) CRC Handbook of laser science and technology,
% *    Vol 4, Pt 2
% * 7. R.S. Krishnan et al.,Thermal Expansion of Crystals, Pergamon Press
% * 8. P. Klocek, Handbook of infrared and optical materials, Marcel Decker,
% *    1991
% * 9. Rai Weiss, electronic log from 5/10/2006
% * 10. Wikipedia online encyclopedia, 2006
% * 11. D.K. Davies, The Generation and Dissipation of Static Charge on
% * dielectrics in a Vacuum, page 29
% * 12. Gretarsson & Harry, Gretarsson thesis
% * 13. Fejer
% * 14. Braginsky
```

```
%% Infrastructure------------------------------------------------------------

ifo.Infrastructure.Length                          = 2999.8;              % m;


%% Detailed model for residual gas noise----------Giancarlo Cella, 12/05/2009-----------------------------
%% Each quantity is a vector which contain the parameter for one kind of
%% molecules, in the following order:
%%
%%                                                    H2O                 H2                N2              O2
% Pa; partial pressures GOAL:
ifo.Infrastructure.ResidualGas.pressures           = [10^-7              10^-7             4*10^-8         10^-8    ];
% kg; Mass of the molecule:
ifo.Infrastructure.ResidualGas.masses              = [3.01289*10^-26  3.34765*10^-27  4.7*10^-26   5.3*10^-26    ];
% m^3; Gas polarizability:
ifo.Infrastructure.ResidualGas.polarizabilities = [1.43319*10^-30  7.81917*10^-31  1.63*10^-30  1.49*10^-30];


%% Physical and other constantMaterialss; All in SI units----------------------

ifo.Constants.E0 = 8.8541878176e-12;               % F/m; Permittivity of Free Space
ifo.Constants.hbar = 1.054572e-34;                 % J-s; (Plancks constant)/(2*pi) ħ
ifo.Constants.c    = 2.99792458e8;                 % m/s; Speed of light in Vacuum
ifo.Constants.G    = 6.67259e-11;                  % m^3/Kg/s^2; Grav. Constant
ifo.Constants.kB   = 1.380658e-23;                 % J/K; Boltzman Constant
ifo.Constants.h = ifo.Constants.hbar*2*pi;         % J-s; Planks constant
ifo.Constants.R = 8.31447215;                      % J/(K*mol); Gas Constant
ifo.Constants.Temp = 290;                          % K; Temperature of the Vacuum
ifo.Constants.yr = 365.2422*86400;                 % sec; Seconds in a year
ifo.Constants.Mpc = ifo.Constants.yr*ifo.Constants.c*3.26e6;     % m

ifo.Constants.MSol = 1.989e30*ifo.Constants.G/ifo.Constants.c^2;% m;
ifo.Constants.g = 9.81;                            % m/s^2; grav. acceleration
ifo.Constants.fs = 20000;                          % Sampling frequency (Hz)


%% Laser----------------------------------------------------------------------
ifo.Laser.Wavelength                               = 1.064e-6;  % m  λ;
ifo.Laser.Power                                    = 125;    % W (exiting from IMC);
```

```
%ifo.Laser.Power                          = 25;    % W (exiting from IMC);

%% Optics ------------------------------------------------------------------
ifo.Optics.SRM.CavityLength               = 24;    % m; ITM to SRM distance
                                                   % (in effect in Section 5.3
                                                   % l is defined as the
                                                   %  BS-2-PR distance
                                                   %  but it doesn't play any effective role because
                                                   %  the phase phi of Eq.63 is
                                                   %  introduced directly in the code

%TBC
ifo.Optics.PhotoDetectorEfficiency  = 1-0.14;    % photo-detection efficiency:
                                                   % it contains all the
                                                   % losses after the SR (OMC,
                                                   % PD, ...). Defined as
                                                   % lambda_PD in Sec.5.3 and
                                                   % computed in Table 5.4
ifo.Optics.Loss                           = 75e-6/2; % average per mirror power loss.
                                                   % the RTL loss L used
                                                   % in Eq.65 is 2x larger
                                                   % Its value is stated in
                                                   % the Eq.71
ifo.Optics.BSLoss   = 1510.0e-6;                   % power loss near beamsplitter
                                                   % see Section 5.4
ifo.Optics.coupling = 0.997;                       % mismatch btwn arms & SRC modes; used to
                                                   % calculate an effective r_srm
                                                   % see Section 5.4

%% Parameter describing thermal lensing ------------------------------------
% The presumably dominant effect of a thermal lens in the ITMs is an increased
% mode mismatch into the SRC, and thus an increased effective loss of the SRC.
% This increase is estimated by calculating the round-trip loss S in the SRC as
% 1-S = |<Psi| exp(i*phi) |Psi>|^2, where
% |Psi> is the beam hitting the ITM and
% phi = P_coat*phi_coat + P_subs*phi_subs
% with phi_coat & phi__subs the specific lensing profiles
% and P_coat & P_subst the power absorbed in coating and substrate
%
% This expression can be expanded to 2nd order and is given by
% S= s_cc P_coat^2 + 2*s_cs*P_coat*P_subst + s_ss*P_subst^2
```

```
% s_cc, s_cs and s_ss where calculated analytically by Phil Wilems (4/2007)

%TBC
ifo.TCS.s_cc=7.024; % Watt^-2
ifo.TCS.s_cs=7.321; % Watt^-2
ifo.TCS.s_ss=7.631; % Watt^-2

% The hardest part to model is how efficient the TCS system is in
% compensating this loss. Thus as a simple Ansatz we define the
% TCS efficiency TCSeff as the reduction in effective power that produces
% a phase distortion. E.g. TCSeff=0.99 means that the compensated distortion
% of 1 Watt absorbed is eqivalent to the uncompensated distortion of 10mWatt.
% The above formula thus becomes:
% S= s_cc P_coat^2 + 2*s_cs*P_coat*P_subst + s_ss*P_subst^2 * (1-TCSeff)^2
%
% To avoid iterative calculation we define TCS.SCRloss = S as an input
% and calculate TCSeff as a bench output.
% TCS.SRCloss is incorporated as an additional loss in the SRC

ifo.TCS.SRCloss=300.0e-6; % V.Fafone March 2012


%% Seismic and Gravity Gradient Parameters--------VIRGO
%
% reviewed (22/03/2012) by M.Punturo, I.Fiori and P.Ruggi
%
% Seismic noise section
%ifo.Seismic.sp=[4e-8 4e-8 0.1 3.4 0.05 0.04 8.0 0.1 5e-3 44 2]; % Quiet seism
%ifo.Seismic.fi=[0.4 0.6 0.6];                                    % Quiet seism
ifo.Seismic.sp=[4e-8 4e-8 12.0 3.5 0.5 0.04 8.0 0.1 5e-3 44 2]; % Noisy seism
ifo.Seismic.fi=[0.4 0.6 1.2];                                    % Noisy seism
%
% common part
ifo.Seismic.Qi=[3 3 3];
ifo.Seismic.PoWi=[-1 -1 1]; %-1 = pole; 1 = zero


% Gravity Gradient section
ifo.Seismic.Rho = 2.3e3;                % kg/m^3; density of the ground near the mirror
ifo.Seismic.Beta = 2*sqrt(pi)/3;        % quiet times beta = 0.35-0.60, noisy times beta = 0.15-1.4
```

```
%% Suspension: SI Units------------------------------------------------------

% This part replaces the orginially used suspension part from Advanced
% LIGO. This set of suspension parameters only works with suspR_AdV.m.
% Proper documentation of this suspension thermal noise model can be found
% in VIR-015A-09 [26].
%
% to update when the payload design will be available
%
% The SA modelling is introduced through its Transfer Function (zero-poles)
% loaded by an external text file generated through Octopus by P.Ruggi
%
ifo.Suspension.SA_pz = importdata('fullSA_TF_v00r01.txt', ' ', 0);
ifo.Suspension.SA_fnorm = 10;    % normalization frequency
ifo.Suspension.SA_Anorm=6.1106e-017;     % TF amplitude at fnorm
%
% last stage section
%
ifo.Suspension.Temp = ifo.Constants.Temp;
ifo.Suspension.VHCoupling.theta = 1e-3;          % vertical-horizontal x-coupling

ifo.Suspension.Silica.Rho     = 2.2e3;           % Kg/m^3;
ifo.Suspension.Silica.C       = 772;             % J/Kg/K;
ifo.Suspension.Silica.K       = 1.38;            % W/m/kg;
ifo.Suspension.Silica.Alpha   = 3.9e-7;          % 1/K;
%ifo.Suspension.Silica.Alpha   = 5.1e-7;           %% USED IN BENCH4
ifo.Suspension.Silica.dlnEdT = 1.52e-4;          % (1/K), dlnE/dT
ifo.Suspension.Silica.Phi     = 4.1e-10;         % from G Harry e-mail to NAR 27April06 dimensionless units
ifo.Suspension.Silica.Y       = 7.2e10;         % Pa; Youngs Modulus
%ifo.Suspension.Silica.Y       = 7.27e10;           % Young's modulus %%USED IN BENCH 4
ifo.Suspension.Silica.Dissdepth = 1.5e-2;        % from G Harry e-mail to NAR 27April06

ifo.Suspension.C85Steel.Rho     =  7900;          % from producer communication (WAGNER spa)
```

```
ifo.Suspension.C85Steel.C      =  502;                  %     " "
ifo.Suspension.C85Steel.K      =  50;                     %      " "
ifo.Suspension.C85Steel.Alpha  =  14e-6;                 %      " "
ifo.Suspension.C85Steel.Y      = 200e9;                %      "  "
% ifo.Suspension.C85Steel.dlnEdT = -2.782e-4;    % WRONG NUMBER, true for maraging!!!
ifo.Suspension.C85Steel.dlnEdT = -1.3e-4                 % value found for UNS G10860, similar composition
ifo.Suspension.C85Steel.Phi    =  1e-4;          % M.Punturo, entry 22436 (20/03/2009) Virgo logbook

ifo.Suspension.Stage(1).Mass = 42;               %Mirror mass in kg;
ifo.Suspension.Stage(2).Mass = 42;               %Mass of refrenece mass
ifo.Suspension.Stage(3).Mass = 100;              %Mass of marionette

ifo.Suspension.Optcyl=1; %Switch 0=cylindrical fibres; 1=optimized fibres
%----------Optimized fibres----------
ifo.Suspension.Stage(1).WireDiameter1 = 800e-6; %diameter of the first head
ifo.Suspension.Stage(1).WireDiameter2 = 400e-6; %diameter of the middle part
ifo.Suspension.Stage(1).WireDiameter3 = 800e-6; %diameter of the second head
ifo.Suspension.Stage(1).Length1 = 0.05;          %length of the first head
ifo.Suspension.Stage(1).Length2 = 0.6;           %length of the middle part
ifo.Suspension.Stage(1).Length3 = 0.05;          %length of the second head
%----------Cylindrical fibres----------
ifo.Suspension.Stage(1).WireDiameter = 400e-6;   %fibre's diameter
ifo.Suspension.Stage(1).Length = 0.7;            %fibre's length
%--------------------------------------------------

ifo.Suspension.Stage(2).WireDiameter = 600e-6;   %RM wire's diameter
ifo.Suspension.Stage(3).WireDiameter = 1.85e-3; %Marionette wire's diameter

ifo.Suspension.Stage(2).Length = 0.7;            %RM wire's length
ifo.Suspension.Stage(3).Length = 1.125;          %Marionette wire's length

ifo.Suspension.Stage(1).NWires = 4;              %Numbers of wires in mirror suspension
ifo.Suspension.Stage(2).NWires = 4;              %Numbers of wires in RM suspension
ifo.Suspension.Stage(3).NWires = 1;              %Numbers of wires in marionette suspension

ifo.Suspension.Stage(1).Qvh=1e20;                %Viscous Q of mirror for horizontal motion
ifo.Suspension.Stage(2).Qvh=1e5;                 %Viscous Q of reference mass for horizontal motion
ifo.Suspension.Stage(3).Qvh=300;                 %Viscous Q of marionette for horizontal motion

ifo.Suspension.Stage(1).Qvv=1e20;                %Viscous Q of mirror for vertical motion
```

```
ifo.Suspension.Stage(2).Qvv=1e5;                    %Viscous Q of reference mass for vertical motion
ifo.Suspension.Stage(3).Qvv=300;                    %Viscous Q of marionette for verical motion

%% Dielectric coating material parameters----------------------------------------

%% high index material: tantala
ifo.Materials.Coating.Yhighn = 140e9;               % E_n1 in Eq.11
ifo.Materials.Coating.Sigmahighn = 0.23;            % sigma_n1 in Eq.11
ifo.Materials.Coating.CVhighn = 2.1e6;              % Specific heat per volume unit
                                                    % used in Eq.27a
                                                    % see Crooks et al[45]
ifo.Materials.Coating.Alphahighn = 3.6e-6;          % 3.6e-6 Fejer et al, 5e-6 from Braginsky
ifo.Materials.Coating.Betahighn = 1.4e-5;           % dn/dT, value Gretarrson (G070161)
%
ifo.Materials.Coating.ThermalDiffusivityhighn = 33; % Fejer et al
ifo.Materials.Coating.Phihighn = 2.3e-4;
ifo.Materials.Coating.Indexhighn = 2.035;     % Measured at LMA, Email from Laurent (5/3/2009)

%% low index material: silica
ifo.Materials.Coating.Ylown = 72e9;
ifo.Materials.Coating.Sigmalown = 0.17;
ifo.Materials.Coating.CVlown = 1.6412e6;            % Crooks et al, Fejer et al
ifo.Materials.Coating.Alphalown = 5.1e-7;           % Fejer et al
ifo.Materials.Coating.Betalown = 8e-6; % dn/dT,  (ref. 14)
ifo.Materials.Coating.ThermalDiffusivitylown = 1.38; % Fejer et al
ifo.Materials.Coating.Philown = 4.0e-5;
ifo.Materials.Coating.Indexlown = 1.458;          % Measured at LMA, Email from Laurent (5/3/2009)

%%Substrate Material parameters----------------------------------------

ifo.Materials.Substrate.c2  = 7.6e-12;              % Coeff of freq depend.
                                                    % term for bulk mechanical loss, 7.15e-12 for Sup2
ifo.Materials.Substrate.MechanicalLossExponent=0.77;    % Exponent for freq dependence of silica loss,
                                                    % 0.822 for Sup2
ifo.Materials.Substrate.Alphas = 5.2e-12;           % Surface loss limit (ref. 12)
ifo.Materials.Substrate.MirrorY     = 7.27e10;      % N/m^2; Youngs modulus (ref. 4)
ifo.Materials.Substrate.MirrorSigma = 0.167;        % Kg/m^3; Poisson ratio (ref. 4)
ifo.Materials.Substrate.MassDensity = 2.2e3;        % Kg/m^3; (ref. 4)
ifo.Materials.Substrate.MassAlpha = 3.9e-7;         % 1/K; thermal expansion coeff. (ref. 4)
ifo.Materials.Substrate.MassCM = 739;               % J/Kg/K; specific heat (ref. 4)
```

```
ifo.Materials.Substrate.MassKappa = 1.38;                % J/m/s/K; thermal conductivity (ref. 4)
%ifo.Materials.Substrate.RefractiveIndex = 1.45;          % mevans 25 Apr 2008
ifo.Materials.Substrate.RefractiveIndex = 1.452;          % Value form Heraeus, Email from Laurent (5/3/2009)
%
% The following data are extracted from the MIR chapter of the AdV TDR [44]
%
%MIRROR GEOMETRY
ifo.Materials.MassRadius = 0.175;                         % m;
ifo.Materials.MassThickness = 0.200;                      % m;

%BEAM GEOMETRY
%ifo.Optics.Curvature.ITM = 1530;                         % ROC of ITM
%ifo.Optics.Curvature.ETM = 1530;                         % ROC of ETM

ifo.Optics.Curvature.ITM = 1420;                          % ROC of ITM
ifo.Optics.Curvature.ETM = 1683;                          % ROC of ETM

%ifo.Optics.Diameter.ITM = .34;                           % Coating diameter
%ifo.Optics.Diameter.ETM = .34;                           % Coating diameter
ifo.Optics.Diameter.ITM = .33;                            % baffle aperture (1 cm less than the
ifo.Optics.Diameter.ETM = .33;                            % Coating diameter) Sec.2.3.3 of [44]

ifo.Optics.SubstrateAbsorption = 0.3e-4;                  % 1/m; bulk absorption coef
ifo.Optics.pcrit = 10;                                    % W; tolerable heating power (factor 1 ATC)

% These are now computed in precompIFO
%ifo.Optics.ITM.BeamRadius = 0.0554;                      % m; 1/e^2 power%radius
%ifo.Optics.ETM.BeamRadius = 0.065;                       % m; 1/e^2 power radius


ifo.Optics.ITM.CoatingAbsorption = 1.0e-6;                % absorption of ITM

%FINESSE, SR-PR PARAMETERS

ifo.Optics.ITM.Transmittance  = 0.014;                    % Transmittance of ITM
ifo.Optics.ETM.Transmittance  = 1e-6;                     % Transmittance of ETM
%ifo.Optics.SRM.Transmittance  = 1;                       % Transmittance of SRM
ifo.Optics.SRM.Transmittance  = 0.20;                     % Transmittance of SRM
ifo.Optics.PRM.Transmittance  = 0.05;                     % PRM trasmittance
                                                          %
```

```
ifo.Optics.SRM.Tunephase = 0.35;


%ifo.Optics.SRM.Tunephase = 0.0;

ifo.Optics.Quadrature.dc = pi/2;


% coating layer optical thicknesses - mevans June 2008
ifo.Optics.ITM.CoatingThicknessLown = 0.308;
ifo.Optics.ITM.CoatingThicknessCap = 0.5;

ifo.Optics.ETM.CoatingThicknessLown = 0.27;
ifo.Optics.ETM.CoatingThicknessCap = 0.5;
```

```
% SRM tuning, NSNS optimized
% used to compute φ in Eq.63
% φ = (π - Tunephase)/2
% SRM tuning, wideband
%
% demod/detection/homodyne phase
% defined as η in Eq.64
%
```

B Brownian Substrate Code

```
function n = subbrownian(f, ifo)
% SUBBROWNIAN - strain noise psd arising from the Brownian thermal noise
% due to mechanical loss in the substrate material

  wITM = ifo.Optics.ITM.BeamRadius;
  wETM = ifo.Optics.ETM.BeamRadius;
  Y = ifo.Materials.Substrate.MirrorY;
  sigma = ifo.Materials.Substrate.MirrorSigma;

  c2 = ifo.Materials.Substrate.c2;
  n = ifo.Materials.Substrate.MechanicalLossExponent;
  alphas = ifo.Materials.Substrate.Alphas;
  L = ifo.Infrastructure.Length;
  kBT = ifo.Constants.kB*ifo.Constants.Temp;

  % Bulk substrate contribution
  phibulk = c2.*f.^n;

  [cITM, aITM] = subbrownianFiniteCorr(ifo, 'ITM');
  [cETM, aETM] = subbrownianFiniteCorr(ifo, 'ETM');
  cbulk = 8 * kBT * (aITM + aETM) .* phibulk ./ (2 * pi * f);

  % Surface loss contribution
  % csurfETM = alphas/(Y*pi*wETM^2);
  % csurfITM = alphas/(Y*pi*wITM^2);

  csurfETM = alphas*(1-2*sigma)/((1-sigma)*Y*pi*wETM^2);
  csurfITM = alphas*(1-2*sigma)/((1-sigma)*Y*pi*wITM^2);
  csurf = 8 * kBT * (csurfITM + csurfETM) ./ (2 * pi * f);

  % account for 2 ITM and 2 ETM, and convert to strain whith 1/L^2
  n = 2 * (csurf + cbulk) / L^2;
```

C Seismic noise generation Code

```
function n = groundSeism(f, ifo)
%
% M. Punturo (14/03/2012)
%
%
% read the seism parameters
%
sp=ifo.Seismic.sp;
fi=ifo.Seismic.fi;
Qi=ifo.Seismic.Qi;
PoWi=ifo.Seismic.PoWi; %-1 = pole; 1 = zero
%
fnorm = 0;
Anorm = 0;
TF = TransFunct(f,fi,Qi,PoWi, fnorm, Anorm);
%
b1=sp(3)./((f.^2-sp(4)).^2+sp(4).*f./sp(5)); % first bump
b2=sp(6)./(1+exp((f-sp(7))./sp(8)));         % second bump
b3 = (sp(9)/sp(11)).*exp(-0.5.*(((f-sp(10))./sp(11)).^2));      % third bump
bumps=b1+b2+b3;
hfslope=abs(TF);
n=sp(1).*hfslope+sp(2).*bumps;
```

D Brownian Substrate Finite Size Correction Code

```
function [cftm, aftm] = subbrownianFiniteCorr(ifo, opticName)
% subbrownianFiniteCorr - Estimate amplitude coefficient of
% mirror thermal noise contribution for finite-size test masses.
%
% [cftm, aftm] = subbrownianFiniteCorr(ifo, opticName)
% cftm = finite mirror correction factor
% aftm = amplitude coefficient for thermal noise:
%        thermal noise contribution to displacement noise is
%         S_x(f) = (8 * kB * T / (2*pi*f)) * Phi(f) * aftm
%
% Equation references to Bondu, et al. Physics Letters A 246 (1998)
% 227-236 (hereafter BHV) and Liu and Thorne gr-qc/0002055 (hereafter LT)

  % get some numbers
  a = ifo.Materials.MassRadius;
  h = ifo.Materials.MassThickness;
  w = ifo.Optics.(opticName).BeamRadius;
  Y = ifo.Materials.Substrate.MirrorY;
  sigma = ifo.Materials.Substrate.MirrorSigma;
  zeta = ifo.Constants.BesselZeros;

  % do the work
  j0m = besselj(0,zeta);
  r0 = w / sqrt(2);                    % LT uses e-folding of power
  km = zeta/a;

  Qm = exp(-2*km*h);                   % LT eq. 35a

  Um = (1-Qm).*(1+Qm)+4*h*km.*Qm;
  Um = Um./((1-Qm).^2-4*(km*h).^2.*Qm); % LT 53 (BHV eq. btwn 29 & 30)

  x = exp(-(zeta*r0/a).^2/4);
  s = sum(x./(zeta.^2.*j0m));          % LT 57

  x2 = x.*x;
  U0 = sum(Um.*x2./(zeta.*j0m.^2));
  U0 = U0*(1-sigma)*(1+sigma)/(pi*a*Y); % LT 56 (BHV eq. 3)
```

```
p0 = 1/(pi*a^2);                        % LT 28
DeltaU = (pi*h^2*p0)^2;
DeltaU = DeltaU + 12*pi*h^2*p0*sigma*s;
DeltaU = DeltaU + 72*(1-sigma)*s^2;
DeltaU = DeltaU*a^2/(6*pi*h^3*Y);       % LT 54

aftm = DeltaU + U0;                     % LT 58 (eq. following BHV 31)

% amplitude coef for infinite TM
%   factored out: (8 * kB * T * Phi) / (2 * pi * f)
aitm = (1 - sigma^2) / (2 * sqrt(2 * pi) * Y * r0); % LT 59

% finite mirror correction
cftm = aftm / aitm;
```

E Brownian Coating parameters feeding Code

```
function n = coatbrownian_ADV(f,ifo)
% COAT - effect of optical coating on Brownian thermal noise
%   returns strain noise power spectrum in 1 / Hz
%
% Added by G Harry 8/3/02 from work by Nakagawa, Gretarsson, et al.
% Expanded to reduce approximation, GMH 8/03
% Modified to return strain noise, PF 4/07
% Modified to accept coating with non-quater-wave layers, mevans 25 Apr 2008

  % Constants
  L = ifo.Infrastructure.Length;

  % compute noise power from one ITM and one ETM
  % hild@star.sr.bham.ac.uk, 30/08/2008
  % Overall thickness of coating layers is given as an input.

  % The values given here are derived from Raffaeles Email (25/08/2008)
  % SbrITM  = getCoatBrownian_ADV(f, ifo, 'ITM', 1.6e-6, 1.15e-6);
  % SbrETM  = getCoatBrownian_ADV(f, ifo, 'ETM', 3.5e-6, 2.5e-6);

  % Values calculated with method from Laurent (Email 06/03/2009)
  % Finesse 888
  %SbrITM  = getCoatBrownian_ADV(f, ifo, 'ITM', 1.82e-6, 1.18e-6);
  %SbrETM  = getCoatBrownian_ADV(f, ifo, 'ETM', 3.83e-6, 2.61e-6);

  % Values calculated with method from Laurent (Email 06/03/2009)
  % Finesse 444
  SbrITM  = getCoatBrownian_ADV(f, ifo, 'ITM', 1.64e-6, 1.05e-6);

  SbrETM  = getCoatBrownian_ADV(f, ifo, 'ETM', 3.83e-6, 2.61e-6);

  %  Uncomment for 'optimized' coatings
  %SbrITM  = getCoatBrownian(f, ifo, 'ITM');
  %SbrETM  = getCoatBrownian(f, ifo, 'ETM');
```

```
n = 2 * (SbrITM + SbrETM) / L^2;

%   figure
%   loglog(f, sqrt(SbrITM)/L, f, sqrt(SbrETM)/L, f, sqrt(n))
%   legend('single ITM', 'single ETM', 'total coating noise')
%   figure
```

F Brownian Coating Code

```
% SbrZ  = getCoatBrownian(f, ifo, opticName)
% SbrZ  = getCoatBrownian(f, ifo, wBeam, dOpt)
%  returns the coating brownian noise for a given collection of
%  coating layers.  The layers are assumed to be alernating low-n
%  high-n layers, with low-n first.
%
% f = frequency vector in Hz
% ifo = parameter struct from IFOmodel.m
%
% opticName = name of the Optic struct to use for wBeam and dOpt
%   wBeam = ifoArg.Optics.(opticName).BeamRadius
%   dOpt = ifoArg.Optics.(opticName).CoatLayerOpticalThickness
%
% wBeam = beam radius (at 1 / e^2 power)
% dOpt = coating layer thickness vector (Nlayer x 1)
%      = the optical thickness, normalized by lambda, of each coating layer.
%
% SbrZ = Brownian noise spectra for one mirror in m^2 / Hz
%
% adapted from bench62
% based on Harry et al., Class Quant Grav 24 (2007) 405-415

function SbrZ = getCoatBrownian_ADV(f, ifo, wBeam, dlown, dhighn)

  % check arguments
  % if nargin < 4
    % wBeam has been used as the name of an optic
    [wBeam, dOpt] = getCoatParFromName(ifo, wBeam);
  % end

  % Constants
  kBT = ifo.Constants.kB * ifo.Constants.Temp;
  lambda = ifo.Laser.Wavelength;

  Ysub = ifo.Materials.Substrate.MirrorY;
  sigmasub = ifo.Materials.Substrate.MirrorSigma;

  Yhighn = ifo.Materials.Coating.Yhighn;
```

```
sigmahighn = ifo.Materials.Coating.Sigmahighn;
phihighn = ifo.Materials.Coating.Phihighn;
nH = ifo.Materials.Coating.Indexhighn;

Ylown = ifo.Materials.Coating.Ylown;
sigmalown = ifo.Materials.Coating.Sigmalown;
philown = ifo.Materials.Coating.Philown;
nL = ifo.Materials.Coating.Indexlown;

% The following lines ar obsulete for Advanced Virgo as we give directly
% the overall thickness of the lown-coating layer and the highn-coating
% layer.
% hild@star.sr.bham.ac.uk,  30/08/2008

% compute thickness of each material in the coating
% dlown = sum(dOpt(1:2:end)) * lambda / nL;
% dhighn = sum(dOpt(2:2:end)) * lambda / nH;
dCoat = dlown + dhighn;

% for debugging, this is a rough but direct estimate
%   Llown = dlown * philown;
%   Lhighn = dhighn * phihighn;
%   Lsum = Llown + Lhighn;
%   Lall = [Lsum, Llown, Lhighn]

%%%%%%%%%%%%%%%%%% this part is directly from bench62 %%%%%%%%%%%%%%%%%%
Yperp = dCoat/(dhighn/Yhighn+dlown/Ylown);
phiperp = Yperp/dCoat*(dlown*philown/Ylown + dhighn*phihighn/Yhighn);
Ypara = 1/dCoat*(Yhighn*dhighn + Ylown*dlown);
phipara = 1/(dCoat*Ypara)*(Ylown*philown*dlown + Yhighn*phihighn*dhighn);

% This is a kludge, the real formula is very complicted but this
% average works really well
sigma1 = 1/2*(sigmahighn+sigmalown);

% This is exact
sigma2 = (sigmahighn*Yhighn*dhighn+sigmalown*Ylown*dlown)/(Yhighn*dhighn+Ylown*dlown);

% Brownian contribution to coating thermal noise, low Poisson ratio limit
% cITM = dITM/(pi*wITM^2)*(Ypara/Ysub^2*phipara+phiperp/Yperp);
```

```
% cETM = dCoat/(pi*wETM^2)*(Ypara/Ysub^2*phipara+phiperp/Yperp);

% Brownian contribution to coating thermal noise, full formula
c = dCoat*(1-sigmasub^2)/(pi*wBeam^2)*((1/(Yperp*(1-sigmasub^2))-...
  2*sigma2^2*Ypara/(Yperp^2*(1-sigmasub^2)*(1-sigma1)))*phiperp +...
  Ypara*sigma2*(1-2*sigmasub)/(Yperp*Ysub*(1-sigma1)*(1-sigmasub))*(phipara-phiperp)+...
  Ypara*(1+sigmasub)*(1-2*sigmasub)^2/(Ysub^2*(1-sigma1^2)*(1-sigmasub))*phipara);

% noise power
SbrZ = 4 * kBT * c ./ (2 * pi * f);
```

G Thermo-elastic Substrate Code

```
function n = subtherm(f, ifo)
% SUBTHERM - noise psd arising from thermoelastic fluctuations in mirror

  wITM = ifo.Optics.ITM.BeamRadius;
  wETM = ifo.Optics.ETM.BeamRadius;
  sigma = ifo.Materials.Substrate.MirrorSigma;

  L = ifo.Infrastructure.Length;
  kBT = ifo.Constants.kB*ifo.Constants.Temp;

  rho = ifo.Materials.Substrate.MassDensity;      %
  kappa = ifo.Materials.Substrate.MassKappa;      % thermal conductivity
  alpha = ifo.Materials.Substrate.MassAlpha;      % thermal expansion
  CM = ifo.Materials.Substrate.MassCM;            % heat capacity @ constant mass
  Temp = ifo.Constants.Temp;                % temperature

  S0 = 8*(1+sigma)^2*kappa*alpha^2*Temp*kBT;      % note kBT has factor Temp
  S0 = S0/(sqrt(2*pi)*(CM*rho)^2);
  SITM = S0/(wITM/sqrt(2))^3;               % LT 18 less factor 1/omega^2
  SETM = S0/(wETM/sqrt(2))^3;               % LT 18 less factor 1/omega^2

  % Corrections for finite test masses:
  SITM = SITM * subthermFiniteCorr(ifo, 'ITM');
  SETM = SETM * subthermFiniteCorr(ifo, 'ETM');

  % 2 mirrors each type, factor omega^2, dimensionless strain
  n = 2*(SITM + SETM)./(2*pi*f*L).^2;
```

H Thermo-elastic Substrate Finite Correction Code

```
function coeff = subthermFiniteCorr(ifo, opticName)
% CFTM20 - finite size test mass correction to noise amplitude coefficient
% (Liu & Thorne gr-qc/0002055 equation 46)
%
% Equation references to Bondu, et al. Physics Letters A 246 (1998)
% 227-236 (hereafter BHV) or Liu and Thorne gr-qc/0002055 (hereafter LT)

  % extract some numbers
  a = ifo.Materials.MassRadius;
  h = ifo.Materials.MassThickness;
  w = ifo.Optics.(opticName).BeamRadius;
  sigma = ifo.Materials.Substrate.MirrorSigma;
  zeta = ifo.Constants.BesselZeros;

  % do the work
  j0m = besselj(0,zeta);
  r0 = w/sqrt(2);                          % LT uses power e-folding
  km = zeta/a;

  Qm = exp(-2*km*h);                       % LT eq. 35a

  pm = exp(-(km*r0).^2/4)./(pi*(a*j0m).^2);      % LT 37

  c0 = 6*(a/h)^2*sum(j0m.*pm./zeta.^2); % LT 32
  c1 = -2*c0/h;                         % LT 32
  p0 = 1/(pi*a^2);                      % LT 28
  c1 = c1 + p0/(2*h);                   % LT 40

  coeff = (1-Qm).*((1-Qm).*(1+Qm)+8*h*km.*Qm);
  coeff = coeff + 4*(h*km).^2.*Qm.*(1+Qm);
  coeff = coeff.*km.*(pm.*j0m).^2.*(1-Qm);
  coeff = coeff./((1-Qm).^2-4*(h*km).^2.*Qm).^2;
  coeff = sum(coeff) + h*c1^2/(1+sigma)^2;
  coeff = coeff*(sqrt(2*pi)*r0)^3*a^2;  % LT 46
```

I Thermo-optical coating core computation Code

```
% [dphi_dT, dphi_TE, dphi_TR, rCoat] = ...
%   getCoatTOPhase(nIn, nOut, nLayer, dOpt, aLayer, bLayer)
%   returns coating reflection phase derivatives wrt temperature
%
% nIn = refractive index of input medium (e.g., vacuum = 1)
% nOut = refractive index of output medium (e.g., SiO2 = 1.45231 @ 1064nm)
% nLayer = refractive index of each layer, ordered input to output (N x 1)
% dOpt   = optical thickness / lambda of each layer
%        = geometrical thickness * refractive index / lambda
% aLayer = change in geometrical thickness with temperature
%        = the effective thermal expansion coeffient of the coating layer
% bLayer = change in refractive index with temperature
%        = dn/dT
%        = dd/dT - n * a
%
% dphi_dT = total thermo-optic phase derivative with respect to temperature
%         = dphi_TE + dphi_TR
% dphi_TE = thermo-elastic phase derivative (dphi / dT)
% dphi_TR = thermo-refractive phase derivative (dphi / dT)
% rCoat = amplitude reflectivity of coating (complex)
%
% Note about aLayer: on a SiO2 substrate,
% a_Ta2O5 ~ 3.5 * alpha_Ta2O5
% a_SiO2 ~ 2.3 * alpha_SiO2
%
% see getCoatTOPos for more information
% (see also T080101)

function [dphi_dT, dphi_TE, dphi_TR, rCoat] = ...
    getCoatTOPhase(nIn, nOut, nLayer, dOpt, aLayer, bLayer)

% vector of all refractive indexs
nAll = [nIn; nLayer(:); nOut];

% reflectivity of each interface
r = (nAll(1:end-1) - nAll(2:end)) ./ (nAll(1:end-1) + nAll(2:end));

% combine reflectivities
```

```
rbar = zeros(size(r));
ephi = zeros(size(r));

ephi(end) = exp(-4i * pi * dOpt(end));
rbar(end) = ephi(end) * r(end);
for n = numel(dOpt):-1:1
  % one-way phase in this layer
  if n > 1
    ephi(n) = exp(-4i * pi * dOpt(n - 1));
  else
    ephi(n) = 1;
  end

  % accumulate reflecitivity
  rbar(n) = ephi(n) * (r(n) + rbar(n + 1)) / (1 + r(n) * rbar(n + 1));
end

% reflectivity derivatives
dr_dphi = zeros(size(dOpt));
for n = numel(dOpt):-1:1
  dr_dphi(n) = -i * rbar(n + 1);
  for m = n:-1:1
    dr_dphi(n) = dr_dphi(n) * ephi(m) * ...
      (1 - r(m).^2) / (1 + r(m) * rbar(m + 1)).^2;
  end
end

% geometrical distances
dGeo = dOpt ./ nLayer;

% phase derivatives
dphi_dd = 4 * pi * imag(dr_dphi / rbar(1));

% thermo-refractive coupling
dphi_TR = sum(dphi_dd .* (bLayer + aLayer .* nLayer) .* dGeo);

% thermo-elastic
dphi_TE = 4 * pi * sum(aLayer .* dGeo);

% total
```

```
dphi_dT = dphi_TR + dphi_TE;

% coating reflectivity
rCoat = rbar(1);
```

J Thermo-optical coating core computation Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gTC = getCoatThickCorr(f, ifo, dOpt, dTE, dTR)
%   finite coating thickness correction
%     Uses correction factor from T080101, "Thick Coating Correction" (Evans)
%
% (see getCoatThermoOptic for example usage)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For comparison in the bTR = 0 limit, the
% equation from Fejer (PRD70, 2004)
% modified so that gFC -> 1 as xi -> 0
%  gTC = (2 ./ (R * xi.^2)) .* (sh - s + R .* (ch - c)) ./ ...
%    (ch + c + 2 * R * sh + R^2 * (ch - c));
% which in the limit of xi << 1 becomes
%  gTC = 1 - xi * (R - 1 / (3 * R));

function gTC = getCoatThickCorr(f, ifo, dOpt, dTE, dTR)

  % parameter extraction
  pS = ifo.Materials.Substrate;
  Cs = pS.MassCM * pS.MassDensity;
  Ks = pS.MassKappa;

  % compute coating average parameters
  [dc, Cc, Kc] = getCoatAvg(ifo, dOpt);

  % R and xi (from T080101, Thick Coating Correction)
  w = 2 * pi * f;
  R = sqrt(Cc * Kc / (Cs * Ks));
  xi = dc * sqrt(2 * w * Cc / Kc);

  % trig functions of xi
  s = sin(xi);
  c = cos(xi);
  sh = sinh(xi);
  ch = cosh(xi);

  % pR and pE (dTR = -\bar{\beta} lambda, dTE = \Delta \bar{\alpha} d)
```

```
    pR = dTR / (dTR + dTE);
    pE = dTE / (dTR + dTE);

    % various parts of gTC
    g0 = 2 * (sh - s) + 2 * R * (ch - c);
    g1 = 8 * sin(xi / 2) .* (R * cosh(xi / 2) + sinh(xi / 2));
    g2 = (1 + R^2) * sh + (1 - R^2) * s + 2 * R * ch;
    gD = (1 + R^2) * ch + (1 - R^2) * c + 2 * R * sh;

    % and finally, the correction factor
    gTC = (pE^2 * g0 + pE * pR * xi .* g1 + pR^2 * xi.^2 .* g2) ./ (R * xi.^2 .* gD);

end
```

K Thermo-elastic Coating Finite Correction Code

```
% Cfsm = getCoatFiniteCorr(ifo, opticName)
% Cfsm = getCoatFiniteCorr(ifo, wBeam, dOpt)
%   finite mirror size correction
%     Uses correction factor from PLA 2003 vol 312 pg 244-255
%     "Thermodynamical fluctuations in optical mirror coatings"
%     by V. B. Braginsky and S. P. Vyatchanin
%      http://arxiv.org/abs/cond-mat/0302617
%
% ifo = parameter struct from IFOmodel.m
% opticName = name of the Optic struct to use for wBeam and dOpt
%   wBeam = ifoArg.Optics.(opticName).BeamRadius
%   dOpt = ifoArg.Optics.(opticName).CoatLayerOpticalThickness
%
% wBeam = beam radius (at 1 / e^2 power)
% dOpt   = optical thickness / lambda of each layer
%        = geometrical thickness * refractive index / lambda
%
% (see getCoatTOPos for example usage)
%
% version 1 by Sam Wald, 2008

function Cfsm = getCoatFiniteCorr(ifo, wBeam, dOpt)

  % check arguments
  if nargin < 3
    % wBeam has been used as the name of an optic
    [wBeam, dOpt] = getCoatParFromName(ifo, wBeam);
  end

  % parameter extraction
  R = ifo.Materials.MassRadius;        %substrate radius
  H = ifo.Materials.MassThickness;     %substrate thickness
  lambda = ifo.Laser.Wavelength;
  zeta = ifo.Constants.BesselZeros;  % zeros of 1st order bessel function (J1)

  pS = ifo.Materials.Substrate;
  pC = ifo.Materials.Coating;
```

```
alphaS = pS.MassAlpha;
C_S = pS.MassCM * pS.MassDensity;
Y_S = pS.MirrorY;
sigS = pS.MirrorSigma;

alphaL = pC.Alphalown;
C_L = pC.CVlown;
Y_L = pC.Ylown;
sigL = pC.Sigmalown;
nL = pC.Indexlown;

alphaH = pC.Alphahighn;
C_H = pC.CVhighn;
Y_H = pC.Yhighn;
sigH = pC.Sigmahighn;
nH = pC.Indexhighn;

% coating sums
dL = lambda * sum(dOpt(1:2:end)) / nL;
dH = lambda * sum(dOpt(2:2:end)) / nH;
dc = dH + dL;

% AVERAGE SPECIFIC HEAT (simple volume average for coating)
Cf = (C_L * dL + C_H * dH) / dc;
Cr = Cf / C_S;

% COATING AVERAGE VALUE X = ALPHAF*(1+POISSONf)/(1-POISSONf) avg
xxL = alphaL * (1 + sigL) / (1 - sigL);
xxH = alphaH * (1 + sigH) / (1 - sigH);
Xf = (xxL * dL + xxH * dH) / dc;
Xr = Xf / alphaS;

% COATING AVERAGE VALUE Y = ALPHAF* YOUNGSF/(1-POISSONF) avg
yyL = alphaL * Y_L / (1 - sigL);
yyH = alphaH * Y_H / (1 - sigH);
Yf = (yyL * dL + yyH * dH) / dc;
Yr = Yf / (alphaS * Y_S);

% COATING AVERAGE VALUE Z = 1/(1-POISSONF) avg
zzL = 1 / (1 - sigL);
```

```
zzH = 1 / (1 - sigH);
Zf = (zzL * dL + zzH * dH) / dc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FINITE SIZE CORRECTION CALCULATION

% beam size parameter used by Braginsky
r0 = wBeam / sqrt(2);

% values of J0 at zeros of J1
j0m = besselj(0, zeta);

% between eq 77 and 78
km = zeta / R;
Qm = exp(-2 * km * H);
pm = exp(-km.^2 * r0^2 / 4) ./ j0m;  % left out factor of pi * R^2 in denominator

% eq 88
Lm = Xr - Zf * (1 + sigS) + (Yr * (1 - 2 * sigS) + Zf - 2 * Cr) * ...
   (1 + sigS) * (1 - Qm).^2 ./ ((1 - Qm).^2 - 4 * km.^2 * H^2 .* Qm);

% eq 90 and 91
S1 = (12 * R^2 / H^2) * sum(pm ./ zeta.^2);
S2 = sum(pm.^2 .* Lm.^2);
P = (Xr - 2 * sigS * Yr - Cr + S1 * (Cr - Yr * (1 - sigS)))^2 + S2;

% eq 60 and 70
LAMBDA = -Cr + (Xr / (1 + sigS) + Yr * (1 - 2 * sigS)) / 2;

% eq 92
Cfsm = sqrt((r0^2 * P) / (2 * R^2 * (1 + sigS)^2 * LAMBDA^2));

end
```

L Suspension Thermal Noise Code

```
function noise=suspR_AdV(f,ifopar)

% Francesco Piergiovanni (piergiovanni@fi.infn.it)
% Paola Puppo (paola.puppo@ego-gw.it)
% Michele Punturo (michele.punturo@pg.infn.it)

%  Suspension thermal noise is obtained by the fluctuation dissipation theorem approach(see  Virgo note VIR-015A-09).
%  A complete elastic computation for the mirror stage is performed (violin modes are taken into account).
%  A low frequency approximation (elastic energy << gravitational energy) is considered for marionette and RF stages.
%  INPUT: frequency (f), list of parameters(ifopar).
%  OUTPUT: suspension thermal noise PSD (noise).

%In this routine, indexes 1 and 2 refer to marionette and RF. Mirror's stage parameter has no index.

ifo=ifopar;

% General parameter
g=ifo.Constants.g;
kb=ifo.Constants.kB;
Temp=ifo.Suspension.Temp;

% Horizontal to Vertical coupling
thetaHV=ifo.Suspension.VHCoupling.theta;

%---------- Mirror pendulum ----------
%Fused Silica thermo-mechanical properties
alpha=ifo.Suspension.Silica.Alpha;
beta=ifo.Suspension.Silica.dlnEdT;
C=ifo.Suspension.Silica.C;
K=ifo.Suspension.Silica.K;
rho=ifo.Suspension.Silica.Rho;
phi0=ifo.Suspension.Silica.Phi ;
ds=ifo.Suspension.Silica.Dissdepth;
E0=ifo.Suspension.Silica.Y;

Optcyl=ifo.Suspension.Optcyl; %Switch cylindrical/optimized fibres

m=ifo.Suspension.Stage(1).Mass;
```

```
N=ifo.Suspension.Stage(1).NWires;

T=m*g/N; %Tension per fibre

w=2*pi*f;

if Optcyl==0,
    L=ifo.Suspension.Stage(1).Length;
    d=ifo.Suspension.Stage(1).WireDiameter;
    r=d/2;
    II=pi/4*r^4;
else
    L1=ifo.Suspension.Stage(1).Length1;
    L2=ifo.Suspension.Stage(1).Length2;
    L3=ifo.Suspension.Stage(1).Length3;
    d1=ifo.Suspension.Stage(1).WireDiameter1;
    d2=ifo.Suspension.Stage(1).WireDiameter2;
    d3=ifo.Suspension.Stage(1).WireDiameter3;
    r1=d1/2;
    r2=d2/2;
    r3=d3/2;
    L=L1+L2+L3;
end



delta=1e-3;



z=zeros(length(f),3000);
x=zeros(length(f),3000);
TotEnergy=zeros(1,length(f));
Energy=zeros(length(f),3000);
phiST=zeros(1,length(f));
phithT=zeros(1,length(f));

for i=1:length(w),
if Optcyl==0; % Cylindrical fibres
    %Surface contribution for circular cross section fibres (horizontal
```

```
    %motion)
    mu=4/r;
    phiST(i)=phi0*(1+mu.*ds);
    S=pi*r^2;
    %Thermoelastic contribution
    Delta=E0*(alpha-beta*T/(S*E0)).^2*Temp/(rho*C);
    tau=0.0737*rho*C*(2*r)^2/K;
    phithT(i)=Delta*w(i)*tau/(1+w(i)^2*tau^2);
    S0=S;
else  % Optimized fibres
    % Optimized fibres routine is called
    [z(i,:) x(i,:) dummy Energy(i,:)]=An_suspn_3(f(i),0,ifo);
    TotEnergy(i)=sum(Energy(i,:));
    phiST(i)=0;
    phithT(i)=0;

    for j=1:length(z(i,:)),
     if z(i,j)<L1,
       r=r1;
     elseif z(i,j)<L1+L2,
       r=r2;
     else
       r=r3;
     end


    mu=4./r;
    phiS=phi0*(1+mu.*ds);
    S=pi*r^2;
    Delta=E0*(alpha-beta*T/(S*E0)).^2*Temp/(rho*C);
    tau=0.0737*rho*C*(2*r)^2/K;

    % Surface contribution and thermoelastic contribution are weighted by
    % the bending energy

    phiST(i)=phiST(i)+phiS*Energy(i,j);
    phithT(i)=phithT(i)+Delta*w(i)*tau/(1+w(i)^2*tau^2)*Energy(i,j);
    end
    phiST(i)=phiST(i)/TotEnergy(i);
    phithT(i)=phithT(i)/TotEnergy(i);
```

```
    S0=pi*(r1^2*L1+r2^2*L2+r3^2*L3)/L;
end

phiT=phiST+phithT;
end
%------------------------Marionette & RF stages ------------------------

% Marionette and RM stages parameter are loaded.
m1=ifo.Suspension.Stage(3).Mass;
m2=ifo.Suspension.Stage(2).Mass;
LL1=ifo.Suspension.Stage(3).Length;
LL2=ifo.Suspension.Stage(2).Length;

dd1=ifo.Suspension.Stage(3).WireDiameter;
dd2=ifo.Suspension.Stage(2).WireDiameter;

N2=ifo.Suspension.Stage(2).NWires;
rho2=ifo.Suspension.C85Steel.Rho;
C2=ifo.Suspension.C85Steel.C;
K2=ifo.Suspension.C85Steel.K;
alpha2=ifo.Suspension.C85Steel.Alpha;
beta2=ifo.Suspension.C85Steel.dlnEdT;
phi2=ifo.Suspension.C85Steel.Phi;
E2=ifo.Suspension.C85Steel.Y;

N1=ifo.Suspension.Stage(3).NWires;
rho1=ifo.Suspension.C85Steel.Rho;
C1=ifo.Suspension.C85Steel.C;
K1=ifo.Suspension.C85Steel.K;
alpha1=ifo.Suspension.C85Steel.Alpha;
beta1=ifo.Suspension.C85Steel.dlnEdT;
phi1=ifo.Suspension.C85Steel.Phi;
E1=ifo.Suspension.C85Steel.Y;

T2=m2*g/N2;
T1=(m1+m2+m)*g/N1;

S1=pi*(dd1/2)^2;
S2=pi*(dd2/2)^2;
```

```
% Thermoelastic contribution
Delta2=E2*(alpha2-beta2*T2/(S2*E2)).^2*Temp/(rho2*C2);
Delta1=E1*(alpha1-beta1*T1/(S1*E1)).^2*Temp/(rho1*C1);

tau1=0.0737*rho1*C1*(dd1)^2/K1;
tau2=0.0737*rho2*C2*(dd2)^2/K2;

for  i=1:length(w),
     phith1(i)=Delta1*w(i)*tau1/(1+w(i)^2*tau1^2);
     phith2(i)=Delta2*w(i)*tau2/(1+w(i)^2*tau2^2);
end
phiT1=phith1+phi1;
phiT2=phith2+phi2;

M1=m+m1+m2;
M2=m2;
I1=pi/4*(dd1/2).^4;
I2=pi/4*(dd2/2).^4;

% Dilution factors for marionette and RM stages
dil1=sqrt(E1*I1*N1/M1/g/LL1^2);
dil2=sqrt(E2*I2*N2/M2/g/LL2^2);

% Gravitational reaction constants
k0p1=M1*g/LL1*(1+dil1);
k0p2=M2*g/LL2*(1+dil2);


w01=sqrt(k0p1/M1);
w02=sqrt(k0p2/M2);

% Horizontal viscous Q's

Q=ifo.Suspension.Stage(1).Qvh;
Q1=ifo.Suspension.Stage(3).Qvh;
Q2=ifo.Suspension.Stage(2).Qvh;

% Total loss angle internal+viscous
phi1tot=dil1*phiT1+w/(Q1*w01);
phi2tot=dil2*phiT2+w/(Q2*w02);
```

```
kp1=k0p1*(1+1i*phi1tot);
kp2=k0p2*(1+1i*phi2tot);

% Reaction constants for mirror stage fused silica fibres VIR-0.15A-09
% sec. 2
for i=1:length(w),
    if Optcyl==0, %Cylindrical fibres
     E=E0*(1+1i*phiT(i));
     lambda=sqrt(1/(2*E*II)*(T+T*sqrt(1+4*E*II*w(i)^2*rho*S/T^2)));
     p=sqrt(1/(2*E*II)*(-T+T*sqrt(1+4*E*II*w(i)^2*rho*S/T^2)));

     Kpend(i)=4*E*II*lambda*p*(lambda^3*cos(p*L)+lambda^2*sin(p*L)*p...
     +p^3*sin(p*L)+lambda*cos(p*L)*p^2)...
      /(-2*lambda*cos(p*L)*p+lambda^2*sin(p*L)-sin(p*L)*p^2);
    else % Optimized fibres
     [z(i,:) x(i,:) Fend]=An_suspn_3(f(i),phiT(i),ifo);
     Kpend(i)=4*Fend/delta;

    end
end

w0=sqrt(real(Kpend(1))/m);
Kpend=real(Kpend)+1i*(imag(Kpend)+w./(w0*Q).*abs(real(Kpend)));%viscous dissipation is added



% Thermal noise is obtained from FDT. The equations of motion are written
% in a martix formalism Ah*Xh=B. The admittance is calculated. VIR-0.15A-09
% sec. 1.1-1.2-1.3
Ah = zeros([3,3,length(f)]);
Ah(1,1,:)=kp1+kp2+Kpend-m1*w.^2;
Ah(1,2,:)=-kp2;
Ah(1,3,:)=-Kpend;
Ah(2,1,:)=-kp2;
Ah(2,2,:)=kp2-m2*w.^2;
Ah(2,3,:)=0;
Ah(3,1,:)=-Kpend;
Ah(3,2,:)=0;
```

```
Ah(3,3,:)=Kpend-m*w.^2;
B=[0  0  1];

for  i=1:length(w),
     Xh(:,i)=linsolve(Ah(:,:,i),B');
end

admitt=1i*w.*Xh(3,:);
noise_h = 4*kb*Temp*real(admitt)./w.^2*4/3000^2; %horizontal suspension thermal noise PSD

% VERTICAL MOTION

%Surface contribution for circular cross section fibres (vertical motion)
mu_v=2/r;
phiv=phi0*(1 + mu_v*ds);


wv1=2*pi*0.4; % 0.4Hz is the resonance frequency of the marionette stage due to the anti-spring system
%Vertical resonance frequency for mirror and RM wires (from Hook's law)
wv2=sqrt(4*E2*S2/LL2/M2);
wv=sqrt(4*E0*S0/L/m);

%Vertical viscous Q's
Qv=ifo.Suspension.Stage(1).Qvv;
Qv1=ifo.Suspension.Stage(3).Qvv;
Qv2=ifo.Suspension.Stage(2).Qvv;

% Total loss angle internal+viscous
phiv=phiv+w/(Qv*wv);
phiv1=phi1+w/(Qv1*wv1);
phiv2=phi2+w/(Qv2*wv2);

% Elastic constants for vertical motion
kv1=wv1^2*M1*(1+1i*phiv1);
kv2=wv2^2*M2*(1+1i*phiv2);
kv=wv^2*m*(1+1i*phiv);


% Thermal noise is obtained from FDT. The equations of motion are written
% in a martix formalism Av*Xv=B. The admittance is calculated. VIR-0.15A-09
```

```
% sec. 1.5
Av = zeros([3,3,length(f)]);
Av(1,1,:)=kv1+kv2+kv-m1*w.^2;
Av(1,2,:)=-kv2;
Av(1,3,:)=-kv;
Av(2,1,:)=-kv2;
Av(2,2,:)=kv2-m2*w.^2;
Av(2,3,:)=0;
Av(3,1,:)=-kv;
Av(3,2,:)=0;
Av(3,3,:)=kv-m*w.^2;
B=[0 0 1];

for i=1:length(w),
    Xv(:,i)=linsolve(Av(:,:,i),B');
end

admitt_v=1i*w.*Xv(3,:);
noise_v    = 4*kb*Temp*real(admitt_v)./w.^2*4/3000^2; %vertical suspension thermal noise PSD
noise_cv = thetaHV^2*noise_v; %vertical coupled to horizontal noise
noise=noise_h+noise_cv; % Total th noise PSD

end
```

M Optimised fibre Code

```
function [z x Fend Energy]=An_suspn_3(f,phi,ifopar)
% INPUT: frequency, loss angle, list of parameter
% OUTPUT: z coordinate along the fibre axis, fibre bending, force at the end of the fibre, bending energy distribution
% In this routine the elastic equation for an optimized fibre (3 segments
% fibres) is analytically solved according with VIR-015A-09 section 2.1.
% For any frequency value, suspR_AdV.m calls this function two times:
% first, to find the energy distribution that allow to compute the energy weighted loss angle
% and then, to find the imaginary reaction costant of the fibre.

ifo=ifopar;

N=ifo.Suspension.Stage(1).NWires; % Number of suspension's fibres

E0=ifo.Suspension.Silica.Y;
E=E0*(1+1i*phi); % Fused Silica Young Modulus {Pa]
rho=ifo.Suspension.Silica.Rho; % FS density [kg/m^3]

g=ifo.Constants.g; % Gravitational field [N/kg]
m=ifo.Suspension.Stage(1).Mass;% Mirror mass [kg]


L1=ifo.Suspension.Stage(1).Length1;
L2=ifo.Suspension.Stage(1).Length2;
L3=ifo.Suspension.Stage(1).Length3;



d1=ifo.Suspension.Stage(1).WireDiameter1;
d2=ifo.Suspension.Stage(1).WireDiameter2;
d3=ifo.Suspension.Stage(1).WireDiameter3;

r1=d1/2;
r2=d2/2;
r3=d3/2;

S1=pi*r1^2;
I1=pi/4*r1^4;
```

```
S2=pi*r2^2;
I2=pi/4*r2^4;
S3=pi*r3^2;
I3=pi/4*r3^4;

T=m*g/N; % Fibre tension
delta=1e-3;% shift at the end of the fibre

w=2*pi*f;

%the solutions for the 3 segments contain 4 integration constants each (see VIR-015A-09
%eq(50)). coeff is a vector that contains the twelve integration constants.
%Twelve equation are needed. They are the 4 boundary conditions and the 8 joining conditions (x, x',force, moment)
%in the diameter discontinuity points.

lambda1=sqrt((T+sqrt(T^2+4*E*I1*w^2*rho*S1))/(2*E*I1));
p1=sqrt((-T+sqrt(T^2+4*E*I1*w^2*rho*S1))/(2*E*I1));

lambda2=sqrt((T+sqrt(T^2+4*E*I2*w^2*rho*S2))/(2*E*I2));
p2=sqrt((-T+sqrt(T^2+4*E*I2*w^2*rho*S2))/(2*E*I2));

lambda3=sqrt((T+sqrt(T^2+4*E*I3*w^2*rho*S3))/(2*E*I3));
p3=sqrt((-T+sqrt(T^2+4*E*I3*w^2*rho*S3))/(2*E*I3));

AA=zeros(12,12);

ex1=exp(-lambda1*L1);
ex3=exp(-lambda3*L3);
sn1=sin(p1*L1);
cn1=cos(p1*L1);
sn2=sin(p2*L2);
cn2=cos(p2*L2);
sn3=sin(p3*L3);
cn3=cos(p3*L3);

% eq1: x1(0)=0
AA(1,:)=[1 ex1 1 0 0 0 0 0 0 0 0 0];

% eq2: x1'(0)=0
AA(2,:)=[-lambda1 lambda1*ex1 0 p1 0 0 0 0 0 0 0 0];
```

```
% eq3: x1(L1)=x2(0)
AA(3,:)=[ex1 1 cn1 sn1 -1 0 -1 0 0 0 0 0];

% eq4: x1'(L1)=x2'(0)
AA(4,:)=[-lambda1*ex1 lambda1 -p1*sn1 p1*cn1 lambda2 0 0 -p2 0 0 0 0];

% eq5: F1(L1)=F2(0)
AA(5,:)=[I1*(-lambda1^3*ex1) I1*lambda1^3 I1*p1^3*sn1 I1*(-p1^3*cn1) I2*lambda2^3 0 0 I2*p2^3 0 0 0 0];

% eq6: M1(L1)=M2(0)
AA(6,:)=[I1*lambda1^2*ex1 I1*lambda1^2 I1*(-p1^2*cn1) I1*(-p1^2*sn1) -I2*lambda2^2 0 I2*p2^2 0 0 0 0 0];

% eq7: M2(L2)=M3(0)
AA(7,:)=[0 0 0 0 0 I2*lambda2^2 -I2*p2^2*cn2 -I2*p2^2*sn2 -I3*lambda3^2 -I3*lambda3^2*ex3 I3*p3^2 0 ];

% eq8: F2(L2)=F3(0)
AA(8,:)=[0 0 0 0 0 I2*lambda2^3 I2*p2^3*sn2 I2*(-p2^3*cn2) I3*lambda3^3 -I3*lambda3^3*ex3 0 I3*p3^3];

% eq9: x2'(L2)=x3'(0)
AA(9,:)=[0 0 0 0 0 lambda2 -p2*sn2 p2*cn2 lambda3 -lambda3*ex3 0 -p3];

% eq10: x2(L2)=x3(0)
AA(10,:)=[0 0 0 0 0 1 cn2 sn2 -1 -ex3 -1 0];

% eq11: x3'(L3)=0
AA(11,:)=[0 0 0 0 0 0 0 0 -lambda3*ex3 lambda3 -p3*sn3 p3*cn3];

% eq12: x3(L3)'=delta
AA(12,:)=[0 0 0 0 0 0 0 0 ex3 1 cn3 sn3];

%Force
BB=[0 0 0 0 0 0 0 0 0 0 0 0 delta];

coef=linsolve(AA,BB');

% first fibre's segment
z1=linspace(0,L1,1000);

x1 = coef(1)*exp(-lambda1*z1)+coef(2)*exp(-lambda1*(L1-z1))...
```

```
    +coef(3)*cos(p1*z1)+coef(4)*sin(p1*z1);
x12 = coef(1)*lambda1^2*exp(-lambda1*z1)+coef(2)*lambda1^2*exp(-lambda1*(L1-z1))...
    +coef(3)*(-p1^2)*cos(p1*z1)+coef(4)*(-p1^2)*sin(p1*z1);

Energy1=0.5*E*I1*x12.^2; %bending energy E=1/2 E I x''^2

% second fibre's segment
z2=linspace(0,L2,1000);
x2 = coef(5)*exp(-lambda2*z2)+coef(6)*exp(-lambda2*(L2-z2))...
    +coef(7)*cos(p2*z2)+coef(8)*sin(p2*z2);
x22 = coef(5)*lambda2^2*exp(-lambda2*z2)+coef(6)*lambda2^2*exp(-lambda2*(L2-z2))...
    +coef(7)*(-p2^2)*cos(p2*z2)+coef(8)*(-p2^2)*sin(p2*z2);
Energy2=0.5*E*I2*x22.^2;%bending energy E=1/2 E I x''^2

% third fibre's segment
z3=linspace(0,L3,1000);
x3 = coef(9)*exp(-lambda3*z3)+coef(10)*exp(-lambda3*(L3-z3))...
    +coef(11)*cos(p3*z3)+coef(12)*sin(p3*z3);
x32 = coef(9)*lambda3^2*exp(-lambda3*z3)+coef(10)*lambda3^2*exp(-lambda3*(L3-z3))...
    +coef(11)*(-p3^2)*cos(p3*z3)+coef(12)*(-p3^2)*sin(p3*z3);

x33 = coef(9)*(-lambda3^3)*ex3+coef(10)*lambda3^3+coef(11)*p3^3*sn3+coef(12)*(-p3^3)*cn3;
Energy3=0.5*E*I3*x32.^2; %bending energy E=1/2 E I x''^2


z=[z1 z2+z1(end) z3+z2(end)+z1(end)];
x=[x1 x2 x3];

Fend=-E*I3*x33;%Force at the end of the fibre
Energy=[Energy1 Energy2 Energy3]; %Energy distribution along the fibre
end
```

N Residual Gas Code

```
function n = gas_adv(f, ifo, mol)
% The following function models the noise spectrum caused by the passage of
% residual gas molecules through the laser beams in the arms of the
% interferometer.  The method used here is presented by Rainer Weiss,
% Micheal E. Zucker, and Stanley E. Whitcomb in their paper Optical
% Pathlength Noise in Sensitive Interferometers Due to Residual Gas.
% Added to Bench by Zhigang Pan, Summer 2006
% Cleaned up by PF, Apr 07; eliminated numerical integration and
% substituted first order expansion of exp, to speed it up

% This function is based on the standard 'gas.m' function form gwinc v1.
% The modifications have been performed by Giancarlo Cella. 12/05/2009.

 L = ifo.Infrastructure.Length;
 Lambda = ifo.Laser.Wavelength;
 k = ifo.Constants.kB;
 T = ifo.Constants.Temp;
 P = ifo.Infrastructure.ResidualGas.pressures(mol);          % Pressure inside the vacuum
 M = ifo.Infrastructure.ResidualGas.masses(mol) ;
 R1 =ifo.Optics.Curvature.ITM;                               % Radius of curvature of ITM
 R2 =ifo.Optics.Curvature.ETM;                               % Radius of curvature of ETM
 alpha = ifo.Infrastructure.ResidualGas.polarizabilities(mol);


 rho = (P)/(k*T);                                            % number density of Gas
 v0 = sqrt((2*k*T)/(M));                                      % Mean speed of Gas

 g1 = 1-(L/R1);                                      % first resonator g-parameter of the ITM
 g2 = 1-(L/R2);                                      % second resonator g-parameter of the ETM
 waist = (L*Lambda)/pi;
 waist = waist*sqrt(((g1*g2)*(1-g1*g2))/((g1+g2-2*g1*g2)^2));
 waist = sqrt(waist);                                % gaussian beam waist size
 zr = pi*waist^2/Lambda;                             % Rayleigh range
 z1 = -((g2*(1-g1))/(g1+g2-2*g1*g2))*L;              % location of ITM relative to the waist
 z2 = ((g1*(1-g2))/(g1+g2-2*g1*g2))*L;               % location of ETM relative to the waist

% The exponential of Eq. 1 of P940008 is expanded to first order; this
% can be integrated analytically
```

```
zint = log(z2 + sqrt(z2^2 + zr^2)) - log(z1 + sqrt(z1^2 + zr^2));
zint = zint*zr/waist;
zint = zint - 2*pi*L*f/v0;
% optical path length for one arm
zint = zint*((4*rho*(2*pi*alpha)^2)/v0);
% eliminate any negative values due to first order approx.
zint(zint < 0) = 0;
% account for both arms & turn into strain noise power
n = 2*zint/L^2;


% disp(sprintf('waist: %8.5f,\t z1: %8.3f \t z2: %8.3f',waist,z1,z2));
```

O Quantum (optical read–out) noise

```
function n = shotrad(f, ifo)
% Quantum noise model
% All references to Buonanno & Chen PRD 64 042006 (2001) (hereafter BnC)
% Updated to include losses DEC 2006 Kirk McKenzie using BnC notation

% f                                          % Signal Freq. [Hz]
lambda  = ifo.Laser.Wavelength;              % Laser Wavelength [m]
hbar    = ifo.Constants.hbar;                % Plancks Constant [Js]
c       = ifo.Constants.c;                   % SOL [m/s]
Omega   = 2*pi*f;                            % [BnC, table 1] Signal angular frequency [rads/s]
omega_0 = 2*pi*c/lambda;                     % [BnC, table 1] Laser angular frequency [rads/s]

%------------------------------------------------------------------
L       = ifo.Infrastructure.Length;         % Length of arm cavities [m]
l       = ifo.Optics.SRM.CavityLength;       % SRC Length [m]
T       = ifo.Optics.ITM.Transmittance;      % ITM Transmittance [Power]
m       = ifo.Materials.MirrorMass;          % Mirror mass [kg]

%------------------------------------------------------------------
bsloss  = ifo.Optics.BSLoss;                 % BS Loss [Power]
mismatch = 1 - ifo.Optics.coupling;          % Mismatch
mismatch = mismatch + ifo.TCS.SRCloss;       % Mismatch

% BSloss + mismatch has been incorporated into a SRC Loss
lambda_SR = mismatch + bsloss;               % SR cavity loss [Power]
lambda_PD = 1 - ifo.Optics.PhotoDetectorEfficiency; % Loss in PD Process [Power]

tau     = sqrt(ifo.Optics.SRM.Transmittance); % SRM Transmittance [amplitude]
rho     = sqrt(1 - tau^2 - lambda_SR);       % SRM Reflectivity [amplitude]

%------------------------------------------------------------------
ds      = ifo.Optics.SRM.Tunephase;          % SRC Detunning
eta     = ifo.Optics.Quadrature.dc;          % Homodyne Readout phase

phi     = (pi-ds)/2;                         % [BnC, between 2.14 & 2.15] SR Detuning
Phi     = mod(l*Omega/c,2*pi);               % [BnC, between 2.14 & 2.15] SRC one pass phase shift

%------------------------------------------------------------------
```

```
lambda_arm = ifo.Optics.Loss*2;                 % [BnC, after 5.2] Round Trip loss in arm [Power]
gamma_ac = T*c/(4*L);                           % [KLMTV-PRD2001] Arm cavity half bandwidth [1/s]
epsilon = lambda_arm/(2*gamma_ac*L/c);          % [BnC, after 5.2] Loss coefficent for arm cavity
Epsilon = 2*epsilon./(1+(Omega/gamma_ac).^2);   % [BnC, 5.3] Loss Parameter

I_0     = ifo.gwinc.pbs;                        % [BnC, Table 1] Power at BS (Power*prfactor) [W]
I_SQL   = (m*L^2*gamma_ac^4)/(4*omega_0);       % [BnC, 2.14] Power to reach free mass SQL
Kappa   = 2*((I_0/I_SQL)*gamma_ac^4)./...
          (Omega.^2.*(gamma_ac^2+Omega.^2));    % [BnC 2.13] Effective Radiation Pressure Coupling
beta    = atan(Omega./gamma_ac);                % [BnC, after 2.11] Phase shift of GW SB in arm
h_SQL   = sqrt(8*hbar./(m*(Omega*L).^2));       % [BnC, 2.12] SQL Strain

%-------------------------------------------------------------------
% Coefficients [BnC, Equations 5.8 to 5.12]
%-------------------------------------------------------------------
C11_L   = sqrt(1-lambda_PD) * ( (1+rho^2) * ( cos(2*phi) + Kappa/2 * sin(2*phi) ) -...
          2*rho*cos(2*beta) - 1/4*epsilon * ( -2 * (1+exp(2*i*beta)).^2 * rho + 4 * (1+rho^2) *...
          cos(beta).^2*cos(2*phi) + ( 3+exp(i*2*beta) ) .* Kappa * (1+rho.^2) * sin(2*phi) ) + ...
        lambda_SR * ( exp(2*i*beta)*rho-1/2 * (1+rho^2) * ( cos(2*phi)+Kappa/2 * sin(2*phi) ) ) );

C22_L   = C11_L;

C12_L   = sqrt(1-lambda_PD) * tau^2 * ( - ( sin(2*phi) + Kappa*sin(phi).^2 )+...
 1/2*epsilon*sin(phi) * ( (3+exp(2*i*beta)) .* Kappa * sin(phi) + 4*cos(beta) .^2 * cos(phi)) + ...
 1/2*lambda_SR * ( sin(2*phi)+Kappa*sin(phi).^2) );

C21_L   = sqrt(1-lambda_PD) * tau^2 * ( (sin(2*phi)-Kappa*cos(phi).^2 ) + ...
     1/2*epsilon*cos(phi) * ( (3+exp(2*i*beta) ).*Kappa*sin(phi) - 4*cos(beta).^2*sin(phi) ) + ...
     1/2*lambda_SR * ( -sin(2*phi) + Kappa*cos(phi).^2) );

%-------------------------------------------------------------------
D1_L    = sqrt(1-lambda_PD) * ( - (1+rho*exp(2*i*beta) ) * sin(phi) + ...
        1/4*epsilon * ( 3+rho+2*rho*exp(4*i*beta) + exp(2*i*beta)*(1+5*rho) ) * sin(phi)+ ...
        1/2*lambda_SR * exp(2*i*beta) * rho * sin(phi) );

D2_L    = sqrt(1-lambda_PD) * ( - (-1+rho*exp(2*i*beta) ) * cos(phi) + ...
        1/4*epsilon * ( -3+rho+2*rho*exp(4*i*beta) + exp(2*i*beta) * (-1+5*rho) ) * cos(phi)+ ...
        1/2*lambda_SR * exp(2*i*beta) * rho * cos(phi) );

%-------------------------------------------------------------------
```

```
P11     = 1/2*sqrt(1-lambda_PD) * sqrt(lambda_SR) * tau *...
          ( -2*rho*exp(2*i*beta)+2*cos(2*phi)+Kappa*sin(2*phi) );
P22     = P11;
P12     = -sqrt(1-lambda_PD) * sqrt(lambda_SR)*tau*sin(phi)*(2*cos(phi)+Kappa*sin(phi) );
P21     =  sqrt(1-lambda_PD) * sqrt(lambda_SR)*tau*cos(phi)*(2*sin(phi)-Kappa*cos(phi) );

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Q11     = sqrt(lambda_PD) *...
          ( exp(-2*i*beta)+rho^2*exp(2*i*beta)-rho*(2*cos(2*phi)+Kappa*sin(2*phi)) + ...
          1/2*epsilon*rho * (exp(-2*i*beta)*cos(2*phi)+exp(2*i*beta).*...
                             ( -2*rho-2*rho*cos(2*beta)+cos(2*phi)+Kappa*sin(2*phi) ) + ...
        2*cos(2*phi)+3*Kappa*sin(2*phi))-1/2*lambda_SR*rho *...
            ( 2*rho*exp(2*i*beta)-2*cos(2*phi)-Kappa*sin(2*phi) ) );
Q22     = Q11;
Q12     = 0;
Q21     = 0;

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
N11     = sqrt(1-lambda_PD) * sqrt(epsilon/2)*tau *(Kappa.*(1+rho*exp(2*i*beta))*sin(phi)+...
          2*cos(beta).*(exp(-i*beta)*cos(phi)-rho*exp(i*beta).*(cos(phi)+Kappa*sin(phi))));
N22     = -sqrt(1-lambda_PD)*sqrt(2*epsilon)*tau*(-exp(-i*beta)+rho*exp(i*beta)).*...
           cos(beta)*cos(phi);
N12     = -sqrt(1-lambda_PD)*sqrt(2*epsilon)*tau*(exp(-i*beta)+rho*exp(i*beta)).*...
           cos(beta)*sin(phi);
N21     = sqrt(1-lambda_PD)*sqrt(2*epsilon)*tau*(-Kappa*(1+rho)*cos(phi)+...
          2*cos(beta).*(exp(-i*beta)+rho*exp(i*beta)).*cos(beta)*sin(phi));


%>>>>>>>>   QUANTUM NOISE POWER SPECTRAL DENSITY [BnC, 5.13]   <<<<<<<<<<<<<<<<

n = h_SQL.^2./(2*Kappa*tau^2.*abs(D1_L*sin(eta)+D2_L*cos(eta)).^2).*(...
    abs(C11_L*sin(eta)+C21_L*cos(eta)).^2+...
    abs(C12_L*sin(eta)+C22_L*cos(eta)).^2+abs(P11*sin(eta)+P21*cos(eta)).^2+...
    abs(P12*sin(eta)+P22*cos(eta)).^2+abs(Q11*sin(eta)+Q21*cos(eta)).^2+...
    abs(Q12*sin(eta)+Q22*cos(eta)).^2+abs(N11*sin(eta)+N21*cos(eta)).^2+...
    abs(N12*sin(eta)+N22*cos(eta)).^2);
```

P Miscellanea of computations

```
% ifo = precompIFO(ifo, PRfixed)
%   add precomputed data to the IFO model
%
% To prevent recomputation of these precomputed data, if the
% ifo argument contains ifo.gwinc.PRfixed, and this matches
% the argument PRfixed, no changes are made.
%
% (mevans June 2008)

function ifo = precompIFO(ifo, PRfixed)

  % check PRfixed
  if isfield(ifo, 'gwinc') && isfield(ifo.gwinc, 'PRfixed') && ...
    ifo.gwinc.PRfixed == PRfixed
    return
  end
  ifo.gwinc.PRfixed = PRfixed;

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DERIVED OPTICS VALES
  % Calculate optics' parameters
  ifo.Materials.MirrorMass = ...
    pi*ifo.Materials.MassRadius^2*ifo.Materials.MassThickness;
  ifo.Materials.MirrorMass = ifo.Materials.MirrorMass* ...
    ifo.Materials.Substrate.MassDensity;                    % Kg
  ifo.Optics.ITM.Thickness = ifo.Materials.MassThickness;



  % Use a function to compute beam diameter (this is a good for "near concentric" cavities)
  [waist, ifo.Optics.ITM.BeamRadius, ifo.Optics.ETM.BeamRadius, waistpos] = OSD_ROC_asym(...
      ifo.Infrastructure.Length, ifo.Optics.Curvature.ITM, ifo.Optics.Curvature.ETM, ifo.Laser.Wavelength);

  % Uncomment if you want to see the calculated beam parameters.
  % disp(sprintf(['The arm cavity waist is %8.5f and is at %8.2f'], waist, waistpos));

  % Calculate clipping losses from finite mirror coating diameter (RLW May
  % 2010)
  clippingloss.ITM = exp(-2*(ifo.Optics.Diameter.ITM/2/ifo.Optics.ITM.BeamRadius).^2);
```

```
clippingloss.ETM = exp(-2*(ifo.Optics.Diameter.ETM/2/ifo.Optics.ETM.BeamRadius).^2);
ifo.Optics.Loss = ifo.Optics.Loss + (clippingloss.ITM + clippingloss.ETM)/2;


% coating layer optical thicknesses - mevans 2 May 2008
ifo.Optics.ITM.CoatLayerOpticalThickness = getCoatDopt(ifo, 'ITM');
ifo.Optics.ETM.CoatLayerOpticalThickness = getCoatDopt(ifo, 'ETM');

% compute power on BS
[pbs, finesse, prfactor, Tpr] = BSPower(ifo, PRfixed);
ifo.gwinc.pbs = pbs;
ifo.gwinc.finesse = finesse;
ifo.gwinc.prfactor = prfactor;
ifo.Optics.PRM.Transmittance = Tpr;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LOAD SAVED DATA
% precompute bessels zeros for finite mirror corrections
global besselzeros;
if isempty(besselzeros)
    % load saved values, or just compute them
    try
        load besselzeros
    catch
        besselzeros = besselzero(1, 300, 1);
    end
end
ifo.Constants.BesselZeros = besselzeros;

% load seismic info
global darmseis_f darmseis_x;
if isempty(darmseis_f) || isempty(darmseis_x)
    load seismic
end
ifo.Seismic.darmseis_f = darmseis_f;
ifo.Seismic.darmseis_x = darmseis_x;



  function [w0, w1, w2, z0] = OSD_ROC_asym(L,Rc1,Rc2,lambda)
```

```
Rc1 = -Rc1;                    % giving the input mirror a negative curvature


z1 = (Rc2*L - L^2)/(Rc1-Rc2+2*L);
z2 = L + z1;

zr = sqrt((Rc1-z1)*z1);
w0 = sqrt(zr*lambda/pi);
w1 = w0 * sqrt(1 +(z1/zr)^2);
w2 = w0 * sqrt(1 +(z2/zr)^2);
z0 = - z1;
```