ENS | PSL

Department of CHEMISTRY

Internship report M1 – 2022
Supporting Information

Brownian Motion near a Soft Surface
Supporting Information

Yilin YE

SUPERVISORS: Yacine AMAROUCHENE, David DEAN, Thomas SALEZ - *Univ. Bordeaux, CNRS, Laboratoire Ondes et Matière d'Aquitaine, UMR 5798, F-33400, Talence, France*

Review: Langevin equation

Friction is not enough

The force $\mathbf{F}$ required to move this particle with the velocity $\mathbf{v}$ in the fluid is $\mathbf{F} = \zeta \mathbf{v}$, with ζ as the friction coefficient. Stokes worked on that in 1850s, finding $\zeta = 6\pi\eta r$. Thus, the equation of motion ($\gamma = \zeta/m$) reads:

$$m\dot{v} = -\zeta v \quad \rightarrow \quad \dot{v} = -\frac{\zeta}{m} v \quad \rightarrow \quad \dot{v} = -\gamma v \quad \rightarrow \quad v(t) = v(0)e^{-\gamma t}$$

This result cannot explain Brown mechanism, since :

- $v(t) \rightarrow 0$ if $t \rightarrow \infty$, but particles would not stop;
- $\langle v^2 \rangle = \langle v^2(0) \rangle e^{-2\gamma t} \rightarrow 0$ if $t \rightarrow \infty$, but it should be $\frac{k_B T}{m}$;

Langevin equation is introduced with the random force δF:

$$m\frac{dv}{dt} = -\zeta v + \delta F$$

where the first term refers to systematic put of the environment influence, while the second term refers to the random put. As for δF, we pose that it shows

- random impacts with solvent molecules;
- very sudden effect, no correlation in space and in time $\langle \delta F(t) \rangle = 0$
- $\langle \delta F(t) \delta F(t') \rangle = 2B\delta(|t' - t|)$, furnished as Gaussian white noise

Solution of Langevin equation

Introduce the Laplace transform:

$$\tilde{f}(s) = \int_0^{+\infty} f(t)e^{-st}dt \qquad \tilde{f}'(s) = s\tilde{f}(s) - f(0)$$

then we will use that to solve the Langevin equation

$$\frac{dv}{dt} = -\gamma v + \frac{\delta F}{m} \qquad \Rightarrow \qquad s\tilde{v}(s) - v(0) = -\gamma v(s) + \frac{\delta F(s)}{m} \qquad \Rightarrow \qquad \tilde{v}(s) = \frac{v(0)}{s+\gamma} + \frac{\delta \tilde{F}(s)}{m(s+\gamma)}$$

We do the inverse Laplace transform according to $\mathscr{L}^{-1}\left\{\tilde{F}(s)\tilde{G}(s)\right\} \to \int_0^t F(t-\tau)G(\tau)d\tau$, obtaining

$$v(t) = v(0)e^{-\gamma t} + \int_0^t dt' \frac{\delta F(t')}{m} \exp\left[-\gamma(t-t')\right]$$

We verify that $\langle v(t)\rangle$ turns to 0, while $t \to \infty$

$$\langle v(t)\rangle = \langle v(0)\rangle e^{-\gamma t} + \int_0^t dt' \frac{\langle\delta F(t')\rangle}{m} \exp\left[-\gamma(t-t')\right] = \langle v(0)\rangle e^{-\gamma t} \to 0$$

As for $\langle v^2(t)\rangle$, we calculate as below:

$$\begin{aligned}
\langle v^2(t)\rangle &= \langle v^2(0)\rangle e^{-2\gamma t} + 2e^{-\gamma t}\int_0^t dt' \frac{\langle\delta F(t)v(0)\rangle}{m} e^{-\gamma(t-t')} + \int_0^t dt_1 \int_0^t dt_2 e^{-\gamma(t-t_1)} e^{-\gamma(t-t_2)} \frac{\langle\delta F(t_1)\delta F(t_2)\rangle}{m^2} \\
&= \langle v^2(0)\rangle e^{-2\gamma t} + \int_0^t dt_1 e^{-2\gamma(t-t_1)} \frac{2B}{m^2} = \langle v^2(0)\rangle e^{-2\gamma t} - \frac{1}{2\gamma}(e^{-2\gamma t}-1) \times \frac{2B}{m^2} \\
&= \langle v^2(0)\rangle e^{-2\gamma t} + \frac{B}{\zeta m}(1-e^{-2\gamma t})
\end{aligned}$$

At long time limit, it is supposed that $\langle v^2(t)\rangle \to \frac{k_B T}{m}$, where k_B is the Boltzmann constant, T is the temperature. Now we have $\langle v^2(t)\rangle \to \frac{B}{\zeta m}$, leading to $B = k_B T\zeta$. Then, we have $\langle\delta F(0)\delta F(t)\rangle = 2k_B T\zeta\delta(t)$. Hence we obtain the fluctuation dissipation theorem:

$$\zeta = \frac{1}{k_B T}\int_0^\infty \langle\delta F(0)\delta F(t)\rangle dt$$

Kubo and Stokes-Einstein relationship

Next, we are interested in MSD and the diffusion coefficient. Consider the following function of v under a given temperature T:

$$\frac{1}{2}m\langle v^2(0)\rangle = \frac{k_B T}{2} \qquad \Rightarrow \qquad \langle v(0)v(t)\rangle = \langle v^2(0)\rangle e^{-\gamma t} = \frac{k_B T}{m}e^{-\gamma t}$$

Thus, we can compute the displacement by:

$$\Delta x(t) = \int_0^t v(\tau_1)d\tau_1 \qquad \Rightarrow \qquad \langle\Delta x^2(t)\rangle = \left\langle \int_0^t d\tau_1 \int_0^t d\tau_2 v(\tau_1)v(\tau_2)\right\rangle$$

with its derivative as:

$$\frac{d}{dt}\langle\Delta x^2(t)\rangle = 2\int_0^t d\tau\langle v(0)v(\tau)\rangle = 2\int_0^t dt'\langle v^2(0)\rangle e^{-\gamma t'} = \frac{2\langle v^2(0)\rangle}{\gamma}\left(1-e^{-\gamma t}\right) = \frac{2k_B T}{\zeta}\left(1-e^{-\gamma t}\right)$$

We integrate a second time, leading to

$$\begin{aligned}
\langle\Delta x^2(t)\rangle &= \langle\Delta x^2(0)\rangle + \frac{2k_B T}{\zeta}\left[t + \frac{1}{\gamma}\left(e^{-\gamma t}-1\right)\right] = 0 + \frac{2k_B T}{\zeta}\left[t + \frac{1}{\gamma}\left(e^{-\gamma t}-1\right)\right] \\
&\xrightarrow{t\ll 1/\gamma} \frac{2k_B T}{\zeta}\left[t + \frac{1}{\gamma}(1-\gamma t)\right] = \frac{k_B T\gamma}{\zeta}t^2 = \frac{k_B T}{m}t^2 \\
&\xrightarrow{t\gg 1/\gamma} \frac{2k_B T}{\zeta}(t+\frac{1}{\gamma}) = \frac{2k_B T}{\zeta}t
\end{aligned}$$

At short time, $\langle\Delta x^2\rangle \propto t^2$, refers to ballistic regime. At long time, $\langle\Delta x^2\rangle \propto t$, refers to diffusive regime.

We can also recover Einstein's relation:

$$\lim_{t\to\infty}\frac{\langle\Delta x^2\rangle}{2t} = \frac{k_B T}{\zeta} = D$$

where D refers to the diffusion coefficient. Or we express that by $\langle v(0)v(t)\rangle$:

$$\int_0^\infty \langle v(0)v(t)\rangle dt = \frac{k_B T}{\zeta} = D$$

which is called Kubo relation.

Towards Fokker-Planck equation

In reality, the molecule is always moving under a potential, with the force $\vec{F} = -\vec{\nabla}U$. Therefore, we modify the Langevin equation as

$$\frac{dp}{dt} = -\zeta\frac{p}{m} - U'(x) + \delta F$$

In practice, it would be very complicated to solve the equation. For example, one issue is that $\langle F(x)\rangle$; it can only be solved in very specific case. An alternative consists in taking a probabilistic picture, and we can adopt a stochastic approach, leading to Fokker-Planck equation.

There are two common choices of discretization: the Itô and the Stratonovich conventions. We employ the following discretization of the Langevin equation:

$$\frac{x_{t+\Delta} - x_t}{\Delta} = -V'(x_t) + \xi_t$$

with an associated discretization of the correlations:

$$\langle f[x(t)]\rangle \to \langle f(x_t)\rangle \qquad \langle f[x(t)]\,\xi(t)\rangle \to \langle f(x_t)\xi_t\rangle \qquad \langle f[x(t)]\,\dot{x}(t)\rangle \to \left\langle f(x_t)\frac{x_{t+\Delta}-x_t}{\Delta}\right\rangle$$

which leads to **Itô's chain rule**:

$$\frac{d}{dt}\langle f[x(t)]\rangle = \left\langle f'[x(t)]\frac{dx}{dt}\right\rangle + T\langle f''[x(t)]\rangle$$

In one spatial dimension x, for an Itô process driven by the standard Wiener process W_t and described by the stochastic differential equation (SDE)

$$dX_t = \mu(X_t,t)dt + \sigma(X_t,t)dW_t$$

with drift $\mu(X_t,t)$ and diffusion coefficient $D(X_t,t) = \sigma^2(X_t,t)/2$, the Fokker-Planck equation for the probability density $p(x,t)$ of the random variable X_t is

$$\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}[\mu(x,t)p(x,t)] + \frac{\partial^2}{\partial x^2}[D(x,t)p(x,t)]$$

Let $\mathbb{P}(x,t)$ be the probability density function to find a particle in $[x, x+dx]$ at time t, and let x satisfy:

$$\dot{x}(t) = -V'(x) + \xi(t)$$

if f is a function, we have:

$$\frac{d}{dt}\langle f[x(t)]\rangle = \frac{d}{dt}\int \mathbb{P}(x,t)f(x)dx = \int \frac{\partial \mathbb{P}(x,t)}{\partial t}f(x)dx$$

but using Itô's chain rule:

$$\frac{d}{dt}\langle f[x(t)]\rangle = \left\langle f'[x(t)]\frac{dx}{dt}\right\rangle + T\langle f''[x(t)]\rangle$$

with Langevin's equation

$$\frac{d}{dt}\langle f[x(t)]\rangle = \langle f'[x(t)]\{-V'[x(t)] + \xi(t)\}\rangle + T\langle f''[x(t)]\rangle$$

since $\langle f'[x(t)]\,\xi(t)\rangle = 0$, we have

$$\frac{d}{dt}\langle f[x(t)]\rangle = \int\left[\frac{df(x)}{dx}\left(-\frac{dV(x)}{dx}\right) + T\frac{d^2f(x)}{dx^2}\right]\mathbb{P}(x,t)dx$$

performing an integration by parts, and using that $\mathbb{P}(x,t)$ is a probability density vanishing at $x \to \infty$:

$$\int \frac{\partial \mathbb{P}(x,t)}{\partial t}f(x)dx = \int \frac{\partial}{\partial x}\left[\frac{dV(x)}{dx} + T\frac{\partial}{\partial x}\right]\mathbb{P}(x,t)f(x)dx$$

this is true for any function f, thus

$$\frac{\partial \mathbb{P}(x,t)}{\partial t} = \frac{\partial}{\partial x}\left[\frac{dV(x)}{dx} + T\frac{\partial}{\partial x}\right]\mathbb{P}(x,t)$$

It could be written as $\partial_t\mathbb{P}(x,t) = -H_{FP}\mathbb{P}(x,t)$ with H_{FP} the Fokker-Planck operator shown above.

Theoretical Analysis

Situation of the problem

The equations of motion are shown below [1]

$$\ddot{X}_G+\frac{2\varepsilon\xi}{3}\frac{\dot{X}_G}{\sqrt{\Delta}}+\frac{\kappa\varepsilon\xi}{6}\left[\frac{19}{4}\frac{\dot{\Delta}\dot{X}_G}{\Delta^{7/2}}-\frac{\dot{\Delta}\dot{\Theta}}{\Delta^{7/2}}+\frac{1}{2}\frac{\ddot{\Theta}-\ddot{X}_G}{\Delta^{5/2}}\right]-\sqrt{\frac{\varepsilon}{2}}\sin\alpha=0$$

$$\ddot{\Delta}+\xi\frac{\dot{\Delta}}{\Delta^{3/2}}+\frac{\kappa\xi}{4}\left[21\frac{\dot{\Delta}^2}{\Delta^{9/2}}-\frac{(\dot{\Theta}-\dot{X}_G)^2}{\Delta^{7/2}}-\frac{15}{2}\frac{\ddot{\Delta}}{\Delta^{7/2}}\right]+\cos\alpha=0$$

$$\ddot{\Theta}+\frac{4\varepsilon\xi}{3}\frac{\dot{\Theta}}{\sqrt{\Delta}}+\frac{\kappa\varepsilon\xi}{3}\left[\frac{19}{4}\frac{\dot{\Delta}\dot{\Theta}}{\Delta^{7/2}}-\frac{\dot{\Delta}\dot{X}_G}{\Delta^{7/2}}+\frac{1}{2}\frac{\ddot{X}_G-\ddot{\Theta}}{\Delta^{5/2}}\right]=0$$

with $\alpha=0$ for a plan case. Hence, $\sin\alpha=0$, and $\cos\alpha=1$.

For the convenience, we simplify all coefficients as:

$$\ddot{\Delta}+a_1\frac{\dot{\Delta}}{\Delta^{3/2}}+a_2\frac{\dot{\Delta}^2}{\Delta^{9/2}}+a_3\frac{\dot{\Theta}^2}{\Delta^{7/2}}+a_3\frac{\dot{X}^2}{\Delta^{7/2}}+a_4\frac{\dot{\Theta}\dot{X}}{\Delta^{7/2}}+a_5\frac{\ddot{\Delta}}{\Delta^{7/2}}+a_6=0$$

$$\ddot{X}_G+b_1\frac{\dot{X}}{\sqrt{\Delta}}+b_2\frac{\dot{\Delta}\dot{X}}{\Delta^{7/2}}+b_3\frac{\dot{\Delta}\dot{\Theta}}{\Delta^{7/2}}+b_4\frac{\ddot{\Theta}}{\Delta^{5/2}}+b_5\frac{\ddot{X}_G}{\Delta^{5/2}}+b_6=0$$

$$\ddot{\Theta}+c_1\frac{\dot{\Theta}}{\sqrt{\Delta}}+c_2\frac{\dot{\Delta}\dot{\Theta}}{\Delta^{7/2}}+c_3\frac{\dot{\Delta}\dot{X}}{\Delta^{7/2}}+c_4\frac{\ddot{X}_G}{\Delta^{5/2}}+c_5\frac{\ddot{\Theta}}{\Delta^{5/2}}+c_6=0$$

with coefficients: $a_1=\xi$, $a_2=\frac{21\kappa\xi}{4}$, $a_3=-\frac{\kappa\xi}{4}$, $a_4=\frac{\kappa\xi}{2}$, $a_5=-\frac{15\kappa\xi}{8}$, $a_6=1$; $b_1=\frac{2\varepsilon\xi}{3}$, $b_2=\frac{19\kappa\xi\varepsilon}{24}$, $b_3=-\frac{\kappa\xi\varepsilon}{6}$, $b_4=\frac{\kappa\xi\varepsilon}{12}$, $b_5=-\frac{\kappa\xi\varepsilon}{12}$, $b_6=0$; and $c_1=\frac{4\varepsilon\xi}{3}$, $c_2=\frac{19\kappa\xi\varepsilon}{12}$, $c_3=-\frac{\kappa\xi\varepsilon}{3}$, $c_4=\frac{\kappa\xi\varepsilon}{6}$, $c_5=-\frac{\kappa\xi\varepsilon}{6}$, $c_6=0$. In addition, we write $\ddot{\Delta},\ddot{X}_G,\ddot{\Theta}$ as $\dot{v}_z,\dot{v}_x,\dot{v}_\theta$, $\dot{\Delta},\dot{X}_G,\dot{\Theta}$ as v_z,v_x,v_θ, Δ,X_G,Θ as r_z,r_x,r_θ.

Effective friction matrix

Mass matrix

Consider the following deterministic equation with mass according to the equations of motion mentioned above:

$$m_\alpha\cdot\dot{v}_\alpha=\left[F_{1\alpha}(\mathbf{x})+F_{2\alpha\beta}(\mathbf{x})\cdot\dot{v}_\beta\right]-m_\alpha\cdot\gamma_{\alpha\beta}\cdot v_\beta$$

so we have:

$$F_{1z}=-m_z a_6=-m_z \qquad F_{1x}=-m_x b_6=0 \qquad F_{1\theta}=-m_\theta c_6=0$$

$$F_{2hzz}=-\frac{m_z a_5}{\Delta^{7/2}} \qquad F_{2hzx}=0 \qquad F_{2hz\theta}=0$$

$$F_{2hxz}=0 \qquad F_{2hxx}=-\frac{m_x b_5}{\Delta^{5/2}} \qquad F_{2hx\theta}=-\frac{m_x b_4}{\Delta^{5/2}}$$

$$F_{2h\theta z}=0 \qquad F_{2h\theta x}=-\frac{m_\theta c_4}{\Delta^{5/2}} \qquad F_{2h\theta\theta}=-\frac{m_\theta c_5}{\Delta^{5/2}}$$

Introduce the mass matrix as $M_{\alpha\beta}=\delta_{\alpha\beta}\cdot m_\alpha-F_{2h\alpha\beta}(\mathbf{x})$:

$$M=\begin{pmatrix} m_z-\frac{15\kappa\xi m_z}{8\Delta^{5/2}} & 0 & 0 \\ 0 & m_x-\frac{\kappa\xi\varepsilon m_x}{12\Delta^{5/2}} & \frac{\kappa\xi\varepsilon m_x}{12\Delta^{5/2}} \\ 0 & \frac{\kappa\xi\varepsilon m_\theta}{6\Delta^{5/2}} & m_\theta-\frac{\kappa\xi\varepsilon m_\theta}{6\Delta^{5/2}} \end{pmatrix}$$

with its inverse matrix M^{-1}:

$$M^{-1}=\begin{pmatrix} \frac{1}{m_z-\frac{15\kappa\xi m_z}{8\Delta^{5/2}}} & 0 & 0 \\ 0 & \frac{12\Delta^{5/2}-2\kappa\xi\varepsilon}{12\Delta^{5/2}m_x-3\kappa\xi\varepsilon m_x} & \frac{\kappa\xi\varepsilon}{3m_\theta\left(\kappa\xi\varepsilon-4\Delta^{5/2}\right)} \\ 0 & \frac{2\kappa\xi\varepsilon}{3m_x\left(\kappa\xi\varepsilon-4\Delta^{5/2}\right)} & \frac{12\Delta^{5/2}-\kappa\xi\varepsilon}{12\Delta^{5/2}m_\theta-3\kappa\xi\varepsilon m_\theta} \end{pmatrix}\simeq\begin{pmatrix} \frac{1}{m_z}+\frac{15\kappa\xi}{8\Delta^{5/2}m_z} & 0 & 0 \\ 0 & \frac{1}{m_x}+\frac{\kappa\xi\varepsilon}{12\Delta^{5/2}m_x} & -\frac{\kappa\xi\varepsilon}{12\Delta^{5/2}m_\theta} \\ 0 & -\frac{\kappa\xi\varepsilon}{6\Delta^{5/2}m_x} & \frac{1}{m_\theta}+\frac{\kappa\xi\varepsilon}{6\Delta^{5/2}m_\theta} \end{pmatrix}$$

Fokker-Planck equation for friction matrix

Let $\mathbb{P}(q,t)$ be the probability density function to find a particle in $[q, q+dq]$, as the general coordinate q satisfies

$$\dot{q}(t) = -W'(q) + \xi(t)$$

where $\xi(t)$ refers to the Wiener process. We have the Fokker-Planck equation as

$$\frac{\partial \mathbb{P}(q,t)}{\partial t} = \frac{\partial}{\partial q}\left[\frac{dW(q)}{dq} + T\frac{\partial}{\partial q}\right]\mathbb{P}(q,t)$$

Suppose that $\mathbf{x}, \mathbf{v}$ refer to the position and velocity, respectively. We consider the following deterministic equation

$$d\mathbf{x} = \mathbf{v}dt \qquad\qquad d\mathbf{v} = -\mathbf{U}dt - \nabla\phi(\mathbf{x})dt$$

where $\phi(\mathbf{x})$ is the external potential. We assume that $\mathbf{U}$ are generated by hydrodynamic interactions, which do not however affect the equilibrium Gibbs-Boltzmann distribution shown as

$$P_{eq}(\mathbf{x},\mathbf{v}) = \frac{1}{\bar{Z}}\exp\left(-\frac{\beta\mathbf{v}^2}{2} - \beta\phi(\mathbf{x})\right)$$

Note, $\frac{\partial P}{\partial x_\alpha} = P\left(-\beta\frac{\partial\phi}{\partial x_\alpha}\right)$, $\frac{\partial P}{\partial v_\alpha} = P(-\beta v_\alpha)$, and $\beta^{-1} = k_B T \overset{k_B=1}{\longrightarrow} T$.

Exploit the Fokker-Planck equation on the distribution probability P above as a function of time t.

$$\begin{aligned}
\frac{\partial P}{\partial t} &= \frac{\partial}{\partial v_\alpha}\left[(U_\alpha + \nabla_\alpha\phi)P + T\frac{\partial P}{\partial v_\alpha}\right] + \frac{\partial}{\partial x_\alpha}\left(T\frac{\partial P}{\partial v_\alpha}\right) \\
&= \frac{\partial}{\partial v_\alpha}\left[(U_\alpha + \nabla_\alpha\phi)P + T\frac{\partial P}{\partial v_\beta}\frac{\partial v_\beta}{\partial v_\alpha}\right] + \frac{\partial}{\partial x_\alpha}\left[T\cdot P(-\beta v_\alpha)\right] \\
&= \frac{\partial}{\partial v_\alpha}\left[T\gamma_{\alpha\beta}\cdot\frac{\partial P}{\partial v_\beta} + U_\alpha P + \frac{\partial\phi}{\partial x_\alpha}P\right] - \frac{\partial}{\partial x_\alpha}(v_\alpha P)
\end{aligned}$$

The last two terms would vanish since

$$\frac{\partial}{\partial v_\alpha}\left(\frac{\partial\phi}{\partial x_\alpha}P\right) = \cancel{\left(\frac{\partial}{\partial v_\alpha}\frac{\partial\phi}{\partial x_\alpha}\right)}\cdot P + \frac{\partial\phi}{\partial x_\alpha}\cdot\frac{\partial P}{\partial v_\alpha} = \frac{\partial\phi}{\partial x_\alpha}\cdot P(-\beta v_\alpha)$$

$$\frac{\partial}{\partial x_\alpha}(v_\alpha P) = \cancel{\left(\frac{\partial v_\alpha}{\partial x_\alpha}\right)}P + v_\alpha\left(\frac{\partial P}{\partial x_\alpha}\right) = v_\alpha\cdot P\cdot\left(-\beta\frac{\partial\phi}{\partial x_\alpha}\right)$$

So we have

$$\frac{\partial P}{\partial t} = \frac{\partial}{\partial v_\alpha}\left(T\gamma_{\alpha\beta}\cdot\frac{\partial P}{\partial v_\beta} + U_\alpha P\right) = \frac{\partial}{\partial v_\alpha}\left(-\gamma_{\alpha\beta}\cdot v_\beta P + U_\alpha P\right) \quad \overset{\text{at equilibrium}}{\Rightarrow} \quad U_\alpha = \gamma_{\alpha\beta}\cdot v_\beta$$

We have for small velocities that

$$U_\alpha = \gamma_{\alpha\beta}\cdot v_\beta = \lambda_{\alpha\beta}(\mathbf{x})\cdot v_\beta + \Lambda_{\alpha\beta\gamma}(\mathbf{x})\cdot v_\beta v_\gamma$$

where the term $\lambda_{\alpha\beta}(\mathbf{x})$ is just the friction tensor without any elastic effects. Additional efforts should be taken on the second term by symmetry. We would like to have

$$\gamma_{\alpha\beta} = \lambda_{\alpha\beta} + \gamma_{2\alpha\beta} \qquad \gamma_{2\alpha\beta} = \Gamma_{\alpha\beta\gamma}\cdot v_\gamma$$

Consequently, we have

$$\Gamma_{\alpha\beta\gamma}(\mathbf{x})\cdot v_\beta v_\gamma = \Lambda_{\alpha\beta\gamma}(\mathbf{x})\cdot v_\beta v_\gamma$$

Without loss of generality, we take

$$\Lambda_{\alpha\beta\gamma} = \Lambda_{\alpha\gamma\beta} \qquad \Rightarrow \qquad \Gamma_{\alpha\beta\gamma} + \Gamma_{\alpha\gamma\beta} = 2\Lambda_{\alpha\beta\gamma}$$

In fact, velocity terms on different directions contribute equally for products, so

$$\Lambda_{\alpha\beta\gamma} = \Lambda_{\alpha\gamma\beta}$$

Also, mutual interactions means that terms with v_α contribute equally toward $\gamma_{\alpha\beta}v_\beta$, hence we obtain another constraint

$$\Gamma_{\alpha\beta\gamma} = \Gamma_{\beta\alpha\gamma}$$

Following the format of Langevin equation, $\gamma_{\alpha\beta}$ matrix above only contains terms about first derivatives

$$\begin{aligned}
U_z &= \gamma_{z\beta}v_\beta = a_1\frac{\dot\Delta}{\Delta^{3/2}} + a_2\frac{\dot\Delta^2}{\Delta^{9/2}} + a_3\frac{\dot\Theta^2+\dot X^2}{\Delta^{7/2}} + a_4\frac{\dot\Theta\dot X}{\Delta^{7/2}}\\
U_x &= \gamma_{x\beta}v_\beta = b_1\frac{\dot X}{\sqrt\Delta} + b_2\frac{\dot\Delta\dot X}{\Delta^{7/2}} + b_3\frac{\dot\Delta\dot\Theta}{\Delta^{7/2}}\\
U_\theta &= \gamma_{\theta\beta}v_\beta = c_1\frac{\dot\Theta}{\sqrt\Delta} + c_2\frac{\dot\Delta\dot\Theta}{\Delta^{7/2}} + c_3\frac{\dot\Delta\dot X}{\Delta^{7/2}}
\end{aligned}$$

From this we take the first equation for instance, finding

$$\sum_\alpha \lambda_{z\alpha}v_\alpha = \xi\frac{v_z}{\Delta^{3/2}} \qquad \Rightarrow \qquad \lambda_{zz} = \frac{\xi}{\Delta^{3/2}} \quad \lambda_{zx}=0 \quad \lambda_{z\theta}=0$$

Similarly, we have

$$\sum_\alpha \lambda_{x\alpha}v_\alpha = \frac{2\varepsilon\xi v_x}{3\Delta^{1/2}} \qquad \Rightarrow \qquad \lambda_{xz}=0 \quad \lambda_{xx}=\frac{2\varepsilon\xi}{3\Delta^{1/2}} \quad \lambda_{x\theta}=0$$

$$\sum_\alpha \lambda_{\theta\alpha}v_\alpha = \frac{4\varepsilon\xi v_\theta}{3\Delta^{1/2}} \qquad \Rightarrow \qquad \lambda_{\theta z}=0 \quad \lambda_{\theta x}=0 \quad \lambda_{\theta\theta}=\frac{4\varepsilon\xi}{3\Delta^{1/2}}$$

Consider

$$\sum_{\alpha\beta}\Lambda_{z\alpha\beta}v_\alpha v_\beta = \frac{21\kappa\xi v_z^2}{4\Delta^{9/2}} - \frac{\kappa\xi\left(v_x^2+v_\theta^2\right)}{4\Delta^{7/2}} + \frac{\kappa\xi v_x v_\theta}{2\Delta^{7/2}}$$

which furnishes

$$\Gamma_{zzz} = \frac{21\kappa\xi}{4\Delta^{9/2}} \qquad \Gamma_{zxx} = -\frac{\kappa\xi}{4\Delta^{7/2}} \qquad \Gamma_{z\theta\theta} = -\frac{\kappa\xi}{4\Delta^{7/2}}$$

Again

$$\sum_{\alpha\beta}\Lambda_{x\alpha\beta}v_\alpha v_\beta = \frac{19\kappa\xi\varepsilon v_z v_x}{24\Delta^{7/2}} - \frac{\kappa\xi\varepsilon v_x v_\theta}{6\Delta^{7/2}}$$

we get

$$\Gamma_{xxz} + \Gamma_{xzx} = \frac{19\kappa\xi\varepsilon}{24\Delta^{7/2}}$$

The symmetry $\Gamma_{\alpha\beta\gamma} = \Gamma_{\beta\alpha\gamma}$ now gives

$$\Gamma_{xxz} = \frac{19\kappa\xi\varepsilon}{24\Delta^{7/2}} - \Gamma_{xzx} = \frac{19\kappa\xi\varepsilon}{24\Delta^{7/2}} - \Gamma_{zxx} = \frac{\kappa\xi}{\Delta^{7/2}}\left(\frac{19\varepsilon}{24}+\frac{1}{4}\right)$$

Therefore, we could resolve all coefficients $\lambda_{\alpha\beta}$ and $\Gamma_{\alpha\beta\gamma}$ by this way:

$$\lambda_{\alpha\beta} = \begin{pmatrix} \frac{a_1}{\Delta^{3/2}} & 0 & 0\\ 0 & \frac{b_1}{\sqrt\Delta} & 0\\ 0 & 0 & \frac{c_1}{\sqrt\Delta}\end{pmatrix}$$

$$\Gamma_{z\alpha\beta} = \begin{pmatrix} \frac{a_2}{\Delta^{9/2}} & 0 & 0\\ 0 & \frac{a_3}{\Delta^{7/2}} & \frac{a_4+b_3-c_3}{2\Delta^{7/2}}\\ 0 & \frac{a_4-b_3+c_3}{2\Delta^{7/2}} & \frac{a_3}{\Delta^{7/2}}\end{pmatrix} \quad \Gamma_{x\alpha\beta} = \begin{pmatrix} 0 & \frac{a_3}{\Delta^{7/2}} & \frac{a_4+b_3-c_3}{2\Delta^{7/2}}\\ \frac{b_2-a_3}{\Delta^{7/2}} & 0 & 0\\ \frac{-a_4+b_3+c_3}{2\Delta^{7/2}} & 0 & 0\end{pmatrix} \quad \Gamma_{\theta\alpha\beta} = \begin{pmatrix} 0 & \frac{a_4-b_3+c_3}{2\Delta^{7/2}} & \frac{a_3}{\Delta^{7/2}}\\ \frac{-a_4+b_3+c_3}{2\Delta^{7/2}} & 0 & 0\\ \frac{c_2-a_3}{\Delta^{7/2}} & 0 & 0\end{pmatrix}$$

Combine those coefficients together by $\gamma_{\alpha\beta} = \lambda_{\alpha\beta} + \Gamma_{\alpha\beta\gamma} v_\gamma$, we obtain

$$\gamma_{\alpha\beta} = \begin{pmatrix} \frac{a_1}{\Delta^{3/2}} + \frac{a_2 v_z}{\Delta^{9/2}} & \frac{v_\theta(a_4+b_3-c_3)}{2\Delta^{7/2}} + \frac{a_3 v_x}{\Delta^{7/2}} & \frac{v_x(a_4-b_3+c_3)}{2\Delta^{7/2}} + \frac{a_3 v_\theta}{\Delta^{7/2}} \\ \frac{v_\theta(a_4+b_3-c_3)}{2\Delta^{7/2}} + \frac{a_3 v_x}{\Delta^{7/2}} & \frac{(b_2-a_3)v_z}{\Delta^{7/2}} + \frac{b_1}{\sqrt{\Delta}} & \frac{v_z(-a_4+b_3+c_3)}{2\Delta^{7/2}} \\ \frac{v_x(a_4-b_3+c_3)}{2\Delta^{7/2}} + \frac{a_3 v_\theta}{\Delta^{7/2}} & \frac{v_z(-a_4+b_3+c_3)}{2\Delta^{7/2}} & \frac{(c_2-a_3)v_z}{\Delta^{7/2}} + \frac{c_1}{\sqrt{\Delta}} \end{pmatrix}$$

and its 1-order approximation of κ:

$$\gamma_{\alpha\beta} \simeq \begin{pmatrix} \frac{\xi}{\Delta^{3/2}} + \frac{21\kappa\xi v_z}{4\Delta^{9/2}} & \frac{\kappa\xi((\varepsilon+3)v_\theta - 3v_x)}{12\Delta^{7/2}} & -\frac{\kappa(\xi(3v_\theta+(\varepsilon-3)v_x))}{12\Delta^{7/2}} \\ \frac{\kappa\xi((\varepsilon+3)v_\theta-3v_x)}{12\Delta^{7/2}} & \frac{2\xi\varepsilon}{3\sqrt{\Delta}} + \frac{\kappa\xi(19\varepsilon+6)v_z}{24\Delta^{7/2}} & -\frac{\kappa(\xi(\varepsilon+1)v_z)}{4\Delta^{7/2}} \\ -\frac{\kappa(\xi(3v_\theta+(\varepsilon-3)v_x))}{12\Delta^{7/2}} & -\frac{\kappa(\xi(\varepsilon+1)v_z)}{4\Delta^{7/2}} & \frac{4\xi\varepsilon}{3\sqrt{\Delta}} + \frac{\kappa\xi(19\varepsilon+3)v_z}{12\Delta^{7/2}} \end{pmatrix}$$

Effective friction matrix

Therefore, the deterministic equation turns to

$$m_\alpha \cdot \dot{v}_\alpha - F_{2\alpha\beta}(\mathbf{x}) \cdot \dot{v}_\beta = M_{\alpha\beta} \cdot \dot{v}_\beta = F_{1\alpha}(\mathbf{x}) - m_\alpha \cdot \gamma_{\alpha\beta} \cdot v_\beta \qquad \Rightarrow \qquad \dot{v}_\beta = M^{-1}_{\alpha\beta}\left(F_{1\alpha}(\mathbf{x}) - m_\alpha \cdot \gamma_{\alpha\beta} v_\beta\right)$$

and then we could finally find the effective friction matrix as $\gamma_{\text{eff}} = M^{-1}_{\alpha\beta} \cdot m_\alpha \cdot \gamma_{\alpha\beta} \cdot v_\beta$

$$\gamma_{\text{eff}} = M^{-1}_{\alpha\beta} \cdot \begin{pmatrix} m_Z & 0 & 0 \\ 0 & m_X & 0 \\ 0 & 0 & m_\Theta \end{pmatrix} \cdot \gamma_{\alpha\beta}$$

with elements below:

$$\gamma_{\text{eff},zz} \simeq \frac{\xi}{\Delta^{3/2}} + \kappa\left(\frac{15\xi^2}{8\Delta^4} + \frac{21\xi v_z}{4\Delta^{9/2}}\right)$$

$$\gamma_{\text{eff},xx} \simeq \frac{2\xi\varepsilon}{3\sqrt{\Delta}} + \frac{\kappa\xi\left(4\sqrt{\Delta}\xi\varepsilon^2 + 18v_z + 57\varepsilon v_z\right)}{72\Delta^{7/2}}$$

$$\gamma_{\text{eff},\theta\theta} \simeq \frac{4\xi\varepsilon}{3\sqrt{\Delta}} + \frac{\kappa\xi\left(8\sqrt{\Delta}\xi\varepsilon^2 + 57\varepsilon v_z + 9v_z\right)}{36\Delta^{7/2}}$$

$$\gamma_{\text{eff},xz} = \gamma_{\text{eff},zx} \simeq \frac{\kappa\xi\left((\varepsilon+3)v_\theta - 3v_x\right)}{12\Delta^{7/2}}$$

$$\gamma_{\text{eff},\theta z} = \gamma_{\text{eff},z\theta} \simeq \frac{\kappa\xi\left((3-\varepsilon)v_x - 3v_\theta\right)}{12\Delta^{7/2}}$$

$$\gamma_{\text{eff},\theta x} = \gamma_{\text{eff},x\theta} \simeq -\frac{\kappa\xi\left(16\Delta^3\xi\varepsilon^2 + 36\Delta^{5/2}(\varepsilon+1)v_z\right)}{144\Delta^6}$$

Modified noise correlator amplitude

Suppose $\gamma_{\text{eff}} \simeq \Psi + \kappa\Phi$, and $\gamma^{1/2}_{\text{eff}} \simeq \psi + \kappa\chi$, so

$$\gamma_{\text{eff}} = \gamma^{1/2}_{\text{eff}}\gamma^{1/2}_{\text{eff}} = (\psi + \kappa\chi)(\psi + \kappa\chi) \simeq \psi\psi + \kappa(\psi\chi + \chi\psi) = \Psi + \kappa\Phi$$

As for 0-order matrix, Ψ is a diagonal matrix, and so is ψ.

$$\psi\psi = \Psi \qquad \Rightarrow \qquad \psi_{ij} = \sqrt{\Psi_{ij}}$$

Besides, as a symmetric matrix, non-diagonal elements would only appear in Φ, and then in χ, resulting in $\psi\chi = \chi\psi$.

$$\psi\chi + \chi\psi = \Phi \qquad \Rightarrow \qquad \chi_{ij} = \frac{\Phi_{ij}}{\sqrt{\Psi_{ii}} + \sqrt{\Psi_{jj}}}$$

Suppose $\gamma_{\text{eff}} = \gamma_0 + \gamma_1(\kappa) + \gamma_{1v}(\kappa, v_i)$,

$$\gamma_0 = \begin{pmatrix} \frac{\xi}{\Delta^{3/2}} & 0 & 0 \\ 0 & \frac{2\xi\varepsilon}{3\sqrt{\Delta}} & 0 \\ 0 & 0 & \frac{4\xi\varepsilon}{3\sqrt{\Delta}} \end{pmatrix} \qquad \gamma_1 = \begin{pmatrix} \frac{15\kappa\xi^2}{8\Delta^4} & 0 & 0 \\ 0 & \frac{\kappa\xi^2\varepsilon^2}{18\Delta^3} & -\frac{\kappa\xi^2\varepsilon^2}{9\Delta^3} \\ 0 & -\frac{\kappa\xi^2\varepsilon^2}{9\Delta^3} & \frac{2\kappa\xi^2\varepsilon^2}{9\Delta^3} \end{pmatrix}$$

$$\gamma_{1v} = \begin{pmatrix} \frac{21\kappa\xi v_z}{4\Delta^{9/2}} & \frac{\kappa\xi((\varepsilon+3)v_\theta - 3v_x)}{12\Delta^{7/2}} & \frac{\kappa\xi((3-\varepsilon)v_x - 3v_\theta)}{12\Delta^{7/2}} \\ \frac{\kappa\xi((\varepsilon+3)v_\theta - 3v_x)}{12\Delta^{7/2}} & \frac{\kappa\xi(6+19\varepsilon)v_z}{24\Delta^{7/2}} & -\frac{\kappa\xi(\varepsilon+1)v_z}{4\sqrt{\Delta}} \\ \frac{\kappa\xi((3-\varepsilon)v_x - 3v_\theta)}{12\Delta^{7/2}} & -\frac{\kappa\xi(\varepsilon+1)v_z}{4\sqrt{\Delta}} & \frac{\kappa\xi(19\varepsilon+3)v_z}{12\Delta^{7/2}} \end{pmatrix}$$

Also, we suppose that $\gamma_{1v,ij} = g_{ij\alpha}v_\alpha$, where $g_{ij\alpha}$ refers to the coefficient of v_α in $\gamma_{1v,ij}$. All non-zero elements are listed below:

$$g_{11z} = \frac{21\kappa\xi}{4\Delta^{9/2}} \qquad g_{12x} = \frac{-\kappa\xi}{4\Delta^{7/2}} \quad g_{12\theta} = \frac{\kappa\xi(\varepsilon+3)}{12\Delta^{7/2}} \quad g_{13x} = \frac{\kappa\xi(3-\varepsilon)}{12\Delta^{7/2}} \quad g_{13\theta} = \frac{-\kappa\xi}{4\Delta^{7/2}}$$
$$g_{21x} = \frac{-\kappa\xi}{4\Delta^{7/2}} \quad g_{21\theta} = \frac{\kappa\xi(\varepsilon+3)}{12\Delta^{7/2}} \qquad g_{22z} = \frac{\kappa\xi(6+19\varepsilon)}{24\Delta^{7/2}} \qquad g_{23z} = -\frac{\kappa\xi(\varepsilon+1)}{4\sqrt{\Delta}}$$
$$g_{31x} = \frac{\kappa\xi(3-\varepsilon)}{12\Delta^{7/2}} \quad g_{31\theta} = \frac{-\kappa\xi}{4\Delta^{7/2}} \qquad g_{32z} = -\frac{\kappa\xi(\varepsilon+1)}{4\sqrt{\Delta}} \qquad g_{33z} = \frac{\kappa\xi(19\varepsilon+3)v_z}{12\Delta^{7/2}}$$

We write $\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_1$, where the former is on 0 order while the latter 1 order. Similarly, $\delta\mathbf{F} = \delta\mathbf{F}_0 + \delta\mathbf{F}_1$, $M = M_0 + M_1$, $M^{-1} = \left(M^{-1}\right)_0 + \left(M^{-1}\right)_1$. We just write $\left(M^{-1}\right)_i$ as M_i^{-1} for the convenience. Note, $M_{1,\alpha\beta}^{-1} \neq 1/M_{1,\alpha\beta}$. Focusing on the 1-order correction v_{i1}, we expand the matrix equation by Einstein sommation convention as

$$(s+\gamma_{i0})\widetilde{v_{i1}} = -\sum_j \gamma_{1,ij}\widetilde{v_{i0}} - \sum_j\sum_k g_{ijk}\widetilde{(v_{j0}\cdot v_{k0})} + M_{0i}^{-1}\widetilde{\delta F_{i1}} + \sum_j M_{1,ij}^{-1}\widetilde{\delta F_{j0}}$$

with the following decomposition towards each term:

$$\mathbf{v}_1 = \mathbf{v}_{gv} + \left(\mathbf{v}_{vv} + \mathbf{v}_{vf} + \mathbf{v}_{fv} + \mathbf{v}_{ff}\right) + \mathbf{v}_{fm} + \mathbf{v}_{mf}$$

$$\widetilde{v_{i,gv}} = \frac{1}{s+\gamma_{i0}}\left(-\sum_j \gamma_{1,ij}\widetilde{v_{j0}}\right) \qquad \widetilde{v_{i,fm}} = \frac{1}{s+\gamma_{i0}}\left(M_{0i}^{-1}\widetilde{\delta F_{i1}}\right) \qquad \widetilde{v_{i,mf}} = \frac{1}{s+\gamma_{i0}}\left(\sum_j M_{1,ij}^{-1}\widetilde{\delta F_{j0}}\right)$$

$$\widetilde{v_{i,vv}} + \widetilde{v_{i,vf}} + \widetilde{v_{i,fv}} + \widetilde{v_{i,ff}} = \frac{1}{s+\gamma_{i0}}\left[-\sum_j\sum_k g_{ijk}\widetilde{(v_{j0}\cdot v_{k0})}\right]$$

We list all solutions along the direction i:

$$v_{i,gv} = \frac{\gamma_{1,ij}}{\gamma_{i0}-\gamma_{j0}}\left[\left(e^{-\gamma_{i0}t} - e^{-\gamma_{j0}t}\right)v_j(0) + \int_0^t \mathrm{d}\tau \frac{\delta F_{j0}(\tau)}{m_j}\left[e^{-\gamma_{i0}(t-\tau)} - e^{-\gamma_{j0}(t-\tau)}\right]\right]$$

$$v_{i,fm} = \int_0^t \mathrm{d}\tau \frac{\delta F_{i1}(\tau)}{m_i} e^{-\gamma_{i0}(t-\tau)} \qquad v_{i,mf} = M_{1,ij}^{-1}\int_0^t \mathrm{d}\tau \delta F_{j0}(\tau) e^{-\gamma_{i0}(t-\tau)}$$

$$v_{i,vv} = -g_{ijk}v_j(0)v_k(0)\cdot\frac{e^{-(\gamma_{j0}+\gamma_{k0})t} - e^{-\gamma_{i0}t}}{\gamma_{i0}-\gamma_{j0}-\gamma_{k0}}$$

$$v_{i,fv} = -g_{ijk}v_k(0)\int_0^t \mathrm{d}\tau \delta F_{j0}(\tau)\frac{e^{-(\gamma_{j0}+\gamma_{k0})(t-\tau)} - e^{-\gamma_{i0}(t-\tau)}}{m_j\left(\gamma_{i0}-\gamma_{j0}-\gamma_{k0}\right)}$$

$$v_{i,vf} = -g_{ijk}v_j(0)\int_0^t \mathrm{d}\tau \delta F_{k0}(\tau)\frac{e^{-(\gamma_{j0}+\gamma_{k0})(t-\tau)} - e^{-\gamma_{i0}(t-\tau)}}{m_k\left(\gamma_{i0}-\gamma_{j0}-\gamma_{k0}\right)}$$

$$v_{i,ff} = -\frac{g_{ijk}}{m_j m_k}\int_0^t \mathrm{d}\tau \int_0^\tau \mathrm{d}x \delta F_{j0}(x) e^{-\gamma_{j0}(\tau-x)}\int_0^\tau \mathrm{d}y \delta F_{k0}(y) e^{-\gamma_{k0}(\tau-y)} e^{-\gamma_{i0}(t-\tau)}$$

As for the modified noise amplitude along z:

$$\langle \delta F_z(\tau_1)\delta F_z(\tau_2)\rangle = 2k_\mathrm{B}Tm_z\delta(\tau_1-\tau_2)\cdot(\gamma_{z0}-\gamma_{1,zz})$$

which is always valid at 1-order correction. Since $\gamma_{z0} = \frac{a_1}{\Delta^{3/2}}$, $\gamma_{1,zz} = -\frac{a_1 a_5}{\Delta^4}$, $M_{1,zz}^{-1} = -\frac{a_5}{\Delta^{5/2}m_z}$, we verify

$$\gamma_{z0} + \gamma_{1,zz} - 2\gamma_{z0}m_z M_{1,zz}^{-1} = \frac{a_1}{\Delta^{3/2}} + \frac{a_5 a_1}{\Delta^4} \equiv \gamma_{z0} - \gamma_{1,zz}$$

Furthermore, we could repeat the same procedure for v_{1x} and $v_{1\theta}$, deriving the modified noise correlator amplitudes K_x and K_θ. There are non-zero non-diagonal elements in γ_1, so we get additional terms shown below:

$$v_{x1}(t) = -v_x(0)\gamma_{1,xx}te^{-\gamma_{x0}t} + \frac{v_\theta(0)\gamma_{1,x\theta}}{\gamma_{x0}-\gamma_{\theta 0}}\left(e^{-\gamma_{x0}t}-e^{-\gamma_{\theta 0}t}\right) - \gamma_{1,xx}\int_0^t \mathrm{d}\tau(t-\tau)e^{-\gamma_{x0}(t-\tau)}\frac{\delta F_{x0}(\tau)}{m_x}$$
$$+\frac{\gamma_{1,x\theta}}{\gamma_{\theta 0}-\gamma_{x0}}\int_0^t \mathrm{d}\tau\left(e^{-\gamma_{\theta 0}(t-\tau)}-e^{-\gamma_{x0}(t-\tau)}\right)\frac{\delta F_{\theta 0}(\tau)}{m_\theta} + \int_0^t \mathrm{d}\tau e^{-\gamma_{x0}(t-\tau)}\left[M^{-1}_{1,xx}\delta F_{x0}(\tau)+M^{-1}_{1,x\theta}\delta F_{\theta 0}(\tau)+\frac{\delta F_{x1}(\tau)}{m_x}\right]$$

$$v_{\theta 1}(t) = -v_\theta(0)\gamma_{1,\theta\theta}te^{-\gamma_{\theta 0}t} + \frac{v_x(0)\gamma_{1,\theta x}}{\gamma_{x0}-\gamma_{\theta 0}}\left(e^{-\gamma_{x0}t}-e^{-\gamma_{\theta 0}t}\right) + \frac{\gamma_{1,\theta x}}{\gamma_{\theta 0}-\gamma_{x0}}\int_0^t \mathrm{d}\tau\left(e^{-\gamma_{\theta 0}(t-\tau)}-e^{-\gamma_{x0}(t-\tau)}\right)\frac{\delta F_{x0}(\tau)}{m_x}$$
$$-\gamma_{1,\theta\theta}\int_0^t \mathrm{d}\tau(t-\tau)e^{-\gamma_{\theta 0}(t-\tau)}\frac{\delta F_{\theta 0}(\tau)}{m_\theta} + \int_0^t \mathrm{d}\tau e^{-\gamma_{\theta 0}(t-\tau)}\left[M^{-1}_{1,\theta x}\delta F_{x0}(\tau)+M^{-1}_{1,\theta\theta}\delta F_{\theta 0}(\tau)+\frac{\delta F_{\theta 1}(\tau)}{m_\theta}\right]$$

Again, we suppose $\langle \delta F_{x0}(\tau_1)\delta F_{x1}(\tau_2)\rangle = K_x\cdot\delta(\tau_1-\tau_2)$, and $\langle \delta F_{\theta 0}(\tau_1)\delta F_{\theta 1}(\tau_2)\rangle = K_\theta\cdot\delta(\tau_1-\tau_2)$ for $\langle v_x^2\rangle$ and $\langle v_\theta^2\rangle$. At long time limit $t\to\infty$, they converge to:

$$\langle v_x^2(t)\rangle = k_\mathrm{B}T\left[\frac{1}{m_x}+2\left(M^{-1}_{1,xx}-\frac{\gamma_{1,xx}}{2m_x\gamma_{x0}}\right)\right]+\frac{K_x}{m_x^2\gamma_{x0}}$$
$$\langle v_\theta^2(t)\rangle = k_\mathrm{B}T\left[\frac{1}{m_\theta}+2\left(M^{-1}_{1,\theta\theta}-\frac{\gamma_{1,\theta\theta}}{2m_\theta\gamma_{x0}}\right)\right]+\frac{K_\theta}{m_\theta^2\gamma_{\theta 0}}$$

Since they should be equal to $\frac{k_\mathrm{B}T}{m_x}$, $\frac{k_\mathrm{B}T}{m_\theta}$, respectively, we get:

$$K_x = k_\mathrm{B}Tm_x\left(\gamma_{1,xx}-2m_xM^{-1}_{1,xx}\gamma_{x0}\right) \quad\Rightarrow\quad \langle\delta F_x(\tau_1)\delta F_x(\tau_2)\rangle = 2k_\mathrm{B}Tm_x\delta(\tau_1-\tau_2)\cdot(\gamma_{x0}-\gamma_{1,xx})$$
$$K_\theta = k_\mathrm{B}Tm_\theta\left(\gamma_{1,\theta\theta}-2m_\theta M^{-1}_{1,\theta\theta}\gamma_{\theta 0}\right) \quad\Rightarrow\quad \langle\delta F_\theta(\tau_1)\delta F_\theta(\tau_2)\rangle = 2k_\mathrm{B}Tm_\theta\delta(\tau_1-\tau_2)\cdot\left(\gamma_{\theta 0}-\gamma_{1,\theta\theta}\right)$$

Numerical Simulations

Discretisation algorithm

We set $N_\mathrm{max}=60000$ for the following numerical simulations. Besides, we introduce another parameter, the time scaling ratio $t_r=1/200$, which means that we would split the time unit into 200 intervals, for the sake of much smooth numerical results. Consider the real time unit less than the typical time of Brownian motion, for example 1ms; so we use $dt=\frac{1\mathrm{ms}\cdot c}{r\sqrt{2\varepsilon}}\cdot t_r$ in the simulation codes.

Dimensionless variables

In order to non-dimensionalize the problem, we follow the variables used in [1]:

$$\delta=\Delta\cdot r\varepsilon \qquad x_G=X_G\cdot r\sqrt{2\varepsilon} \qquad \theta=\Theta\cdot\sqrt{2\varepsilon}$$

Introduce $t=T\cdot r\sqrt{2\varepsilon}/c$, where c is the maximum velocity for free fall particles. So the velocities in reality would be replaced by those shown in our equations of motion:

$$v_\Delta=\frac{v_z}{c}\cdot\sqrt{\frac{2}{\varepsilon}} \qquad v_X=\frac{v_x}{c} \qquad v_\Theta=\frac{v_\theta r}{c}$$

where c is a free fall velocity scale constant

$$c=\sqrt{2gr\rho^*/\rho}$$

and $\rho^*=\rho_\mathrm{sty}-\rho_\mathrm{sol}$, $\rho=\rho_\mathrm{sol}=1.00\ \mathrm{g/cm^3}$, $\rho_\mathrm{sty}=1.06\ \mathrm{g/cm^3}$. Similarly, related to velocities, accelerations and forces would be treated in the same way according to their direction.

$$\dot v_\Delta=\dot v_z\cdot\frac{2r}{c^2} \qquad \dot v_X=\dot v_x\cdot\frac{r\sqrt{2\varepsilon}}{c^2} \qquad \dot v_\Theta=\dot v_\theta\cdot\frac{r^2\sqrt{2\varepsilon}}{c^2}$$

If we write $\widehat{\delta F}$ as the non-dimensional variables, we can find

$$\left\langle\widehat{\delta F_z}(\tau_1)\cdot\widehat{\delta F_z}(\tau_2)\right\rangle=\langle\delta F_z(\tau_1)\cdot\delta F_z(\tau_2)\rangle\times\frac{2}{c^2\varepsilon}$$

$$\left\langle\widehat{\delta F_x}(\tau_1)\cdot\widehat{\delta F_x}(\tau_2)\right\rangle=\langle\delta F_x(\tau_1)\cdot\delta F_x(\tau_2)\rangle\times\frac{1}{c^2}$$

$$\left\langle\widehat{\delta F_\theta}(\tau_1)\cdot\widehat{\delta F_\theta}(\tau_2)\right\rangle=\langle\delta F_\theta(\tau_1)\cdot\delta F_\theta(\tau_2)\rangle\times\frac{r^2}{c^2}$$

Euler-Maruyama method

In Itô calculus, the Euler–Maruyama method is used for the approximate numerical solution of a stochastic differential equation (SDE). Consider the equation

$$dX_t = a(X_t,t)dt + b(X_t,t)dW_t$$

with initial condition $X_0 = x_0$, where W_t stands for the Wiener process, and suppose that we wish to solve this SDE on some interval of time $[0,T]$. Then the Euler-Maruyama approximation to the true solution X is the Markov chain Y defined as follows:

- partition the interval $[0,T]$ into N equal subintervals of width $\Delta t = T/N > 0$: $0 = \tau_0 < \tau_1 < \cdots < \tau_N = T$
- set $Y_0 = x_0$
- recursively define Y_n for $0 \leqslant n \leqslant N-1$ by

$$Y_{n+1} = Y_n + a(Y_n,\tau_n)\Delta t + b(Y_n,\tau_n)\Delta W_n$$

  where the random variables ΔW_n are independent and identically distributed normal random variables with expected value zero and variance Δt.

Discrete-Time Langevin Integration

Consider a Langevin equation with the external force $f(t)$, namely the gravity and the spurious force for us:

$$dv = \frac{f(t)}{m}dt - \gamma v dt + \sqrt{\frac{2\gamma}{\beta m}}dW(t)$$

we exploit the discretisation algorithm based on [2] with the following splitting steps:

$$\begin{aligned}
\mathbf{v}\left(n+\frac{1}{4}\right) &= \sqrt{a}\cdot\mathbf{v}(n) + \left[\frac{1}{\beta}(\mathbf{1}-\mathbf{a})\cdot\mathbf{m}^{-1}\right]^{1/2}\cdot\mathbf{N}^{+}(n)\\
\mathbf{v}\left(n+\frac{1}{2}\right) &= \mathbf{v}\left(n+\frac{1}{4}\right) + \frac{\Delta t}{2}\mathbf{b}\cdot\mathbf{m}^{-1}\cdot\mathbf{f}(n)\\
\mathbf{r}\left(n+\frac{1}{2}\right) &= \mathbf{r}(n) + \frac{\Delta t}{2}\mathbf{b}\cdot\mathbf{v}\left(n+\frac{1}{2}\right)\\
\mathscr{H}(n) &\to \mathscr{H}(n+1)\\
\mathbf{r}(n+1) &= \mathbf{r}\left(n+\frac{1}{2}\right) + \frac{\Delta t}{2}\mathbf{b}\cdot\mathbf{v}\left(n+\frac{1}{2}\right)\\
\mathbf{v}\left(n+\frac{3}{4}\right) &= \mathbf{v}\left(n+\frac{1}{2}\right) + \frac{\Delta t}{2}\mathbf{b}\cdot\mathbf{m}^{-1}\cdot\mathbf{f}(n+1)\\
\mathbf{v}(n+1) &= \sqrt{a}\cdot\mathbf{v}\left(n+\frac{3}{4}\right) + \left[\frac{1}{\beta}(\mathbf{1}-\mathbf{a})\cdot\mathbf{m}^{-1}\right]^{1/2}\cdot\mathbf{N}^{-}(n+1)
\end{aligned}$$

where $a_{ij} = \delta_{ij}\exp(-\gamma_i\Delta t)$, $\mathscr{N}^{\pm}$ are independent normally distributed random variables with zero mean and unit variance, $b_{ij} = \delta_{ij}\sqrt{\frac{2}{\gamma_i\Delta t}\tanh\frac{\gamma_i\Delta t}{2}}$

References

[1] T. Salez, L. Mahadevan, *J. Fluid Mech.* **2015**, *779*, 181-196, **Elastohydrodynamics of a sliding, spinning and sedimenting cylinder near a soft wall**.

[2] D. A. Sivak, J. D. Chodera, and G. E. Crooks, *J. Phys. Chem. B* **2014**, *188(24)*, 6466-6474, **Time step rescaling recovers continuous-time dynamical properties for discrete-time Langevin integration of nonequilibrium systems**.

Simulation codes

Below are the codes we used for the numerical simulations. The first file "para.h" stores common variables, while the second file "BEHD_YYE.f90" is the main program.

Listing 1. para.h

```
real*8 rayon,rhosty,rhosol,mass,mz,mx,mt,grav,kappa,eps,xi,kxi,kxe,temp,k_B,beta,clight,dt
common/para/rayon,rhosty,rhosol,mass,mz,mx,mt,grav,kappa,eps,xi,kxi,kxe,temp,k_B,beta,clight,dt


real*8 noise(3,2),Minv(3,3),gmaeff(3,3,3)
common/calcul/noise,Minv,gmaeff
!! "noise" noise = random force; 3: z/x/  ; 2: Box-Muller method
!! "Minv" effective mass matrix inverse; 3,3: matrix index
!! "gmaeff" effective friction matrix; 3: gamma0/gamma1/gamma1v; 3,3: matrix index
```

Listing 2. BEHD_YYE.f90

```
!!!!! Yilin YE @ Jun. 2022
!!!!! EHD + Brownian motion
!!!!! Based on 1412.0162
!!!!! ONLY ONE VARIABLE   z

!!!!! Here WAS a file based on the technique shown in the article
!!!!! J. Phys. Chem. B 2014, 118, 6466  6474
!!!!! See ""Support Information"" for the Multi-dimensional case

!!!!! Here is a file based on the  Euler Maruyama  method.
!!!!! without different gamma values for frictions and noises.


Program main
    use MATHS
    implicit none
    integer :: Nmax=5000*12 !! Define the maximum number of steps
    !real*8,parameter :: pi=3.14159265358979d0 !! constant
    !complex*16,parameter :: Im=(0.d0,1.0d0)     !! imaginary unit
    integer i,j,k,l,dtmax !! loop variables
    real*8 :: time_begin,time_end !! define variables to record time consumed
    real*8,external :: rectgauss

    include "para.h"
    !real*8 :: p12,v14,v24,v34 !! old intermediate variables.
    !real*8 :: coefa(6),coefb(6),coefc(6) !! coefficients for equations of motion
    real*8 :: masse(3),amplitude(3,3),fext(3),tratio,msdanax,msdanat
    !real*8 :: spuriousforce !! spuriousforce, & 3 components z/x/
    real*8,allocatable :: position(:,:),velocity(:,:),force(:) !! define arrays for position/velocity/force
    real*8,allocatable :: msdx(:),msdt(:),sumx(:),sumt(:),Dcoefx(:),Dcoeft(:)
    !! Must decalre all allocatable variables in advance.
    allocate(position(3,Nmax)); allocate(velocity(3,Nmax)); allocate(force(Nmax))
    allocate(msdx(Nmax)); allocate(msdt(Nmax)); allocate(sumx(Nmax)); allocate(sumt(Nmax));
    allocate(Dcoefx(Nmax)); allocate(Dcoeft(Nmax))

    call cpu_time(time_begin)
    open(unit=31,file="random_number_test.txt")
    open(unit=32,file="positions.txt")
    open(unit=33,file="velocitys.txt")
    open(unit=34,file="forces.txt")
    !open(unit=35,file="inter_Euler_Maruyama.txt")
    open(unit=36,file="Mass_Matrix_Inverse.txt")
    open(unit=371,file="gamma0.txt")
    open(unit=372,file="gamma1.txt")
    open(unit=373,file="gamma1v.txt")
    open(unit=381,file="MSD.txt")

    61 format(2x,'Time Step',15x,'Noise z',15x,'Noise x',15x,'Noise  ')
    62 format(2x,'Time Step',12x,'Position z/  ',12x,'Position x',12x,'Angle  ')
    63 format(2x,'Time Step',12x,'Velocitiy z',15x,'Velocity x',15x,'Velocity  ')
    64 format(2x,'Time Step',12x,'Force z',12x,'Noise z',12x,'Noise x',12x,'Noise  ')
    !65 format(2x,'Time Step',12x,'  coeff',12x,'a=exp(-  . t )',12x,'b=sqrt(tanh(gt2)/gt2)',12x,'M inverse')
    66 format(2x,'Time Step',12x,'M_zz',12x,'M_xx',12x,'M_x ',12x,'M_ x',12x,'M_  ')
    67 format(2x,'Time Step',10x,' v_zz ',10x,' v_xx ',10x,'  v_  ',10x,' v_zx ',10x,'  v_z  ',10x,'  v_x  ')
    68 format(2x,'Time Step',9x,'sum x**2',9x,'< x **2>',9x,'D_x',9x,'sum  **2',9x,'<  **2>',9x,'D_t')
    write(31,61); write(32,62); write(33,63); write(34,64); !write(35,65)
```

```
write(36,66); write(371,67); write(372,67); write(373,67); write(381,68)


71 format(f10.4,3(5X,ES18.8)) !! format for file(31)=random_number_test.txt
72 format(f10.4,3(5X,ES18.8)) !! format for file(32)=position_z.txt
73 format(f10.4,3(5X,ES20.6)) !! format for file(33)=velocity_z.txt
74 format(f10.4,4(5X,ES15.6)) !! format for file(34)=force_z.txt
!75 format(f10.1,4(5X,ES20.10)) !! format for file(35)=intermediates.txt
76 format(f10.4,5(5X,ES15.6)) !! format for file(36)=Mass_Matrix_Inverse.txt
77 format(f8.4,6(5X,ES12.4)) !! format for file(371/372/373)
78 format(f10.4,6(5X,ES10.3)) !! format for file(38-)
99 format('It spent',f8.2,' seconds for the whole program.')


rayon = 1.5d-6; rhosty = 1.06e3; rhosol = 1.00e3; grav = 9.80665d0
mass = pi*rayon**2*rhosty; mz = mass; mx = mass; mt = mass*rayon**2/2; masse(1)=mz; masse(2)=mx; masse(3)=mt
temp = 298.0d0; k_B = 1.38064852d-23; beta = 1.d0/(k_B*temp)
clight = sqrt(2.d0*grav*rayon*(1.0d0-rhosol/rhosty))

kappa = 1.0d-4; eps = 0.1d0; xi = 0.1d0; kxi = kappa*xi; kxe = kappa*xi*eps
!coefa(1)=xi;  coefa(2)=21.d0*kxi/4.d0; coefa(3)=-kxi/4.d0; coefa(4)=kxi/2.d0; coefa(5)=-15.d0*kxi/8.d0; coefa(6
!coefb(1)=2.d0/3.d0*eps*xi;  coefb(2)=19.d0*kxe/24.d0; coefb(3)=-kxe/6.d0; coefb(4)=kxe/12.d0; coefb(5)=-kxe/12.
!coefc(1)=4.d0/3.d0*eps*xi;  coefc(2)=19.d0*kxe/12.d0; coefc(3)=-kxe/3.d0; coefc(4)=kxe/6.0d0; coefc(5)=-kxe/6.0

!! Time gap for each step, t = T * r * sqrt(2*eps) / clight. Here we pose dt ~ T
tratio = 1.0d0 / (20.0d0*10)
dt = 1.0d-3*clight/(rayon*sqrt(2.0d0*eps)) * tratio

!! Initiation
position=0.0d0; velocity=0.0d0; force=0.0d0; amplitude=0.0d0
gmaeff=0.0d0; Minv=0.0d0; fext=0.0d0

position(1,1)=1.0d0; position(2,1)=0.0d0; position(3,1)=0.0d0
!p12=0.0d0; v14=0.0d0; v24=0.0d0; v34=0.0d0

do i=1,Nmax-1
    noise = normaldist(0.0d0,1.0d0,3)
    !do j=1,3
    !    noise(j,1) = noise(j,1)*rectGauss(noise(j,1),2.d0)
    !end do
    write(31,71) i*tratio,noise(1,1),noise(2,1),noise(3,1)

    do j=1,3
        do k=1,3
            do l=1,3
                call updategamma(gmaeff(j,k,l),j,k,l,position(1,i),velocity(1,i),velocity(2,i),velocity(3,i))
            end do
            call updateMinverse(Minv(j,k),position(1,i),j,k)
        end do
        amplitude(j,j) = sqrt(2.0d0*(gmaeff(1,j,j) - gmaeff(2,j,j))/(beta*masse(j)))
    end do


    call updateforce(fext(1),position(1,i))
    velocity(:,i+1) = velocity(:,i) + &
    &fext(:) - MATMUL((gmaeff(1,:,:)+gmaeff(2,:,:)+gmaeff(3,:,:)),velocity(:,i)) + MATMUL(amplitude(:,:),noise(:
    velocity(:,i) = velocity(:,i) / clight
    position(:,i+1) = position(:,i) + velocity(:,i) * dt
    !position(1,i+1) = position(1,1)

    Dcoefx(i) = (gmaeff(1,2,2) - gmaeff(2,2,2))/(beta*mx*gmaeff(1,2,2)**2)
    Dcoeft(i) = (gmaeff(1,3,3) - gmaeff(2,3,3))/(beta*mt*gmaeff(1,3,3)**2)

    write(32,72) i*tratio,position(1,i),position(2,i),position(3,i)
    write(33,73) i*tratio,velocity(1,i),velocity(2,i),velocity(3,i)
    write(34,74) i*tratio,force(i), amplitude(1,1)*noise(1,1), amplitude(2,2)*noise(2,1), amplitude(3,3)*noise(3
    !write(35,75) i*tratio,gamma,intma,intmb,Minv
    write(36,76) i*tratio,Minv(1,1),Minv(2,2),Minv(1,2),Minv(2,1),Minv(2,2)
    write(371,77) i*tratio,gmaeff(1,1,1),gmaeff(1,2,2),gmaeff(1,3,3),gmaeff(1,1,2),gmaeff(1,1,3),gmaeff(1,2,3)
    write(372,77) i*tratio,gmaeff(2,1,1),gmaeff(2,2,2),gmaeff(2,3,3),gmaeff(2,1,2),gmaeff(2,1,3),gmaeff(2,2,3)
```

```
        write(373,77) i*tratio,gmaeff(3,1,1),gmaeff(3,2,2),gmaeff(3,3,3),gmaeff(3,1,2),gmaeff(3,1,3),gmaeff(3,2,3)

    end do


    !! MSD
    dtmax=1000*20
    do i=1,dtmax !!  t ?
        sumx(i) = 0.0d0; sumt(i) = 0.0d0
        do j=1,Nmax-dtmax
            msdx(j) = (position(2,j) - position(2,i+j))**2
            sumx(i) = sumx(i) + msdx(j)

            msdt(j) = (position(3,j) - position(3,i+j))**2
            sumt(i) = sumt(i) + msdt(j)
        end do
        msdanax = 0.0d0; msdanat = 0.0d0
        write(381,78) i*tratio, sumx(i), sumx(i)/j, Dcoefx(i)*i*1.0d0, sumt(i), sumt(i)/j, Dcoeft(i)*i*1.0d0
    end do


    deallocate(velocity); deallocate(position); deallocate(force)
    deallocate(msdx); deallocate(msdt); deallocate(sumx); deallocate(sumt); deallocate(Dcoefx); deallocate(Dcoeft)
    close(31); close(32); close(33); close(34); close(35); close(36); close(371); close(372); close(373); close(381)
    call cpu_time(time_end)
    write(*,99) time_end-time_begin
    !write(*,*) 'clight = ',clight

end Program main


!*** Here is the module to furnish NORMAL DISTRIBUTION, namely the white noises.
!!! Reference: https://www.cxyzjd.com/article/za36mize/78948490
!!! Reference: https://en.wikipedia.org/wiki/Box_Muller_transfo
MODULE MATHS
    implicit none
    real(kind=8),parameter :: pi=4.0d0*atan(1.0d0),twopi=2.0d0*pi
    CONTAINS
    function normaldist(mean,std,n) result(r)
        implicit none
        real(kind=8),intent(in) :: mean,std
        integer,intent(in) :: n
        real(kind=8) :: r(n,2)
        real(kind=8),dimension(n,2) :: zeta
        !call random_seed()
        call random_number(zeta)
        r(:,1) = dsqrt(-2.0d0*log(zeta(:,1)))*cos(twopi*zeta(:,2))
        r(:,2) = dsqrt(-2.0d0*log(zeta(:,1)))*sin(twopi*zeta(:,2))
        r = mean + std * r
    end function normaldist
END MODULE MATHS


!*** Here is the function to calculate the gamma (matrix)
real*8 function gammavalue(no,i,j,z,vz,vx,vt)
    implicit none
    include "para.h"
    integer :: no,i,j !! no: gamma No.?; i/j: matrix index
    real*8 :: z,vz,vx,vt
    real*8 :: zroot
    zroot = sqrt(z)

    select case(no)
    case(1)
        if (i.eq.j) then
            if (i.eq.1) gammavalue = xi/(z*zroot)
            if (i.eq.2) gammavalue = (2.d0*xi*eps)/(3.d0*zroot)
            if (i.eq.3) gammavalue = (4.d0*xi*eps)/(3.d0*zroot)
        else
            gammavalue = 0.d0
```

```
        end if
    case(2)
        if ((i-1)*(j-1).eq.0) gammavalue = 0.0d0
        if ((i.eq.1).and.(j.eq.1)) gammavalue = (15.d0*kxi*xi)/(8.d0*z**4)
        if ((i.eq.2).and.(j.eq.2)) gammavalue = (kxi*xi*eps**2)/(18.d0*z**3)
        if ((i.eq.3).and.(j.eq.3)) gammavalue = (2.d0*kxi*xi*eps**2)/(9.d0*z**3)
        if (i*j.eq.6) gammavalue = -(kxi*xi*eps**2)/(9.d0*z**3)
    case(3)
        if ((i.eq.1).and.(j.eq.1)) gammavalue = (21.d0*kxi*vz)/(4.d0*zroot*z**4)
        if ((i.eq.2).and.(j.eq.2)) gammavalue = (kxi*vz*(6.d0+19.d0*eps))/(24.d0*zroot*z**3)
        if ((i.eq.3).and.(j.eq.3)) gammavalue = (kxi*vz*(3.d0+19.d0*eps))/(12.d0*zroot*z**3)
        if (i*j.eq.2) gammavalue = (kxi*((eps+3.d0)*vt-3.d0*vx))/(12.d0*zroot*z**3)
        if (i*j.eq.3) gammavalue = (kxi*((3.d0-eps)*vx-3.d0*vt))/(12.d0*zroot*z**3)
        if (i*j.eq.6) gammavalue = -(kxi*vz*(eps+1.d0))/(4.d0*zroot)
    end select
    return
end function gammavalue

!*** Here is the function to calculate the force depending on z.
real*8 function forcevalue(z)
    !This function would furnish the force on each direction
    !z: vertical position; vz: velocity vector
    implicit none
    real*8 :: z
    real*8 :: zroot,gzzz,gzxx,gztt,coulombmax
    include "para.h"

    zroot = sqrt(z)
    gzzz = (21.0d0*kxi)/(4.0d0*zroot*z**4)
    gzxx = -(kxi)/(4.0d0*zroot*z**3)
    gztt = -(kxi)/(4.0d0*zroot*z**3)
    coulombmax = 1.0e-15*0.0d0

    forcevalue = - grav*mz*(1-rhosol/rhosty) / clight + (gzzz+gzxx+gztt)/(beta*mz) / clight !+ coulombmax/(z**2)*exp


    return
end function forcevalue

!*** Here is the function to calculate inverse mass matrix
real*8 function Minverse(z,i,j)
    implicit none
    include "para.h"
    real*8 :: z,zroot
    integer :: i,j !! matrix index
    zroot = sqrt(z)

    select case(i)
    case(1)
        if (j.eq.1) then
            Minverse = (1.0d0/mz) * (1.0d0 + (15.0d0*kxi)/(8.0d0*zroot*z**2))
        else
            Minverse = 0.0d0
        end if
    case(2)
        if (j.eq.1) Minverse = 0.0d0
        if (j.eq.2) Minverse = (1.0d0/mx) * (1.0d0 + kxe/(12.0d0*zroot*z**2))
        if (j.eq.3) Minverse = -kxe/(12.0d0*zroot*z**2*mt)
    case(3)
        if (j.eq.1) Minverse = 0.0d0
        if (j.eq.2) Minverse = -kxe/(6.0d0*zroot*z**2*mx)
        if (j.eq.3) Minverse = (1.0d0/mt) * (1.0d0 + kxe/(6.0d0*zroot*z**2*mt))
    end select
    return
end function Minverse


!*** Here is the function to cut off Gauss white noise.
real*8 function rectGauss(num,value)
    implicit none
```

```
    real*8 num,value
    if (abs(value).le.num) then
        rectGauss=1.0d0
    else
        rectGauss=0.0d0
    end if
    return
end function rectGauss


!*** Here is the procedure to update gamma.
subroutine updategamma(valeur,no,i,j,z,vz,vx,vt)
    implicit none
    real*8 :: valeur,z,vz,vx,vt
    integer :: no,i,j
    real*8,external :: gammavalue
    include "para.h"
    valeur = gammavalue(no,i,j,z,vz,vx,vt)
end subroutine


!*** Here is the procedure to update force along z direction.
subroutine updateforce(valeur,z)
    implicit none
    real*8 :: valeur,z
    real*8,external :: forcevalue
    !include "para.h"
    valeur = forcevalue(z)
end subroutine



!*** Here is the procedure to update the inverse mass matrix
subroutine updateMinverse(valeur,z,i,j)
    implicit none
    real*8 :: valeur,z
    integer :: i,j
    real*8,external :: Minverse
    valeur = Minverse(z,i,j)
end subroutine
```