Langevin Equation: Force Auto-Correlation and the Memory Kernel

Ryan Houlihan

November 26, 2013

I The Model

The Langevin Equation is a Stochastic Differential equation used to describe dynamics of a system that can be decomposed into fast and slow degrees of freedom. The Langevin Equation is written as:

$$\ddot{X}(t) + \int_0^t \gamma(t-t')\dot{X}(t')dt' = Z(t)$$

where

- $\ddot{X}(t)$: acceleration of slow Brownian particle (marked particle)
- $\gamma(t-t')$: dissipative memory kernel (friction)
- $Z(t)$: random force (noise)

and the auto correlation of the noise is

$$< Z(t)Z(t') >= C(|t-t'|) = C(\tau)$$

when the force is zero centered and stationary. The memory kernel, $\beta(t)$, and the correlation function, $C(t)$, are related via the fluctuation-dissipation theorem

$$C(t) = \beta^{-1}\gamma(t)$$

For our model we take a single marked particle connected by springs to N smaller particles representing a heat bath. The model problem is defined by the Hamiltonian

$$H(P_N, Q_N, p, q) = \frac{1}{2}P_N^2 + V(Q_N) + \frac{1}{2}\sum_{j=1}^{N}\frac{p_j^2}{m_j} + \frac{1}{2}\sum_{j=1}^{N}k_j(q_j - Q_N)^2$$

where Q_N, P_N are the coordinate and momentum of the distinguished particle and q_j, p_j are the coordinates and momenta of the heat bath particles. The Hamilton equations are then

$$\ddot{Q}_N + V'(Q_N) = \sum_{j=1}^{N} k_j(q_j - Q_N)$$

$$\ddot{q}_j = -\omega_j^2(q_j - Q_n)$$

The intial conditions for this system are

$$Q_N(0) = Q_0\ ,\ \ P_N(0) = P_0$$

$$q_j^0 = Q_0 + \left(\frac{1}{\beta k_j}\right)^{1/2} \xi_j$$

$$p_j^0 = \left(\frac{m_j}{\beta}\right)^{1/2} \eta_j$$

where ξ_j and η_j are normally distributed pseudorandom numbers.

The equations for q_j can be solved in terms of the past history of Q_N and then q_j can be substituted back into the eq. for Q_N which gives us the Langevin equation

$$\ddot{Q}_N + V'(Q_N) + \int_0^t \gamma_N(t-t')Q_N(t')dt' = \beta^{-1/2}Z_N(t)$$

The memory kernel $\gamma_N(t)$ is

$$\gamma_N(t) = \sum_{j=1}^{N} k_j cos(\omega_j t)$$

and the random forcing, $\beta^{-1/2}Z_N$, is

$$\beta^{-1/2}Z_N(t) = \sum_{j=1}^{N} k_j^{1/2}[\xi_j cos(\omega_j t) + \eta_j sin(\omega_j t)]$$

Finally, with our given frequencies and taking expectations with respect to ξ_j and η_j, we find that

$$< Z_N(t+t')Z_N(t) >= \gamma_N(t')$$

which is our fluctuation-dissipation relation for the given system.

For our case we want to choose a memory kernel which decays exponentially. Thus we choose our correlation function to be $C(t) = \Gamma\alpha e^{-\alpha t}$. To do so we choose a set of uniformly distributed ω_j and then set k_j to

$$k_j = \frac{2\alpha}{\pi(\alpha^2 + \omega_j^2)}\Delta\omega$$

where $\omega_j = j\Delta\omega$ and $\Delta\omega = \frac{\zeta}{N}$ where ζ is an α dependent quantity and N is the number of "heat bath" particles. This choice allows γ_N to converge to $\gamma(t) = \Gamma\beta e^{-\alpha|t|}$.

ζ for a given α is chosen by plotting $k(\omega)$ and finding the point where $k(\omega)$ is close enough to zero that all remaining ω values can be ignored. This plot for various values of α is shown in (*Fig. 1*).

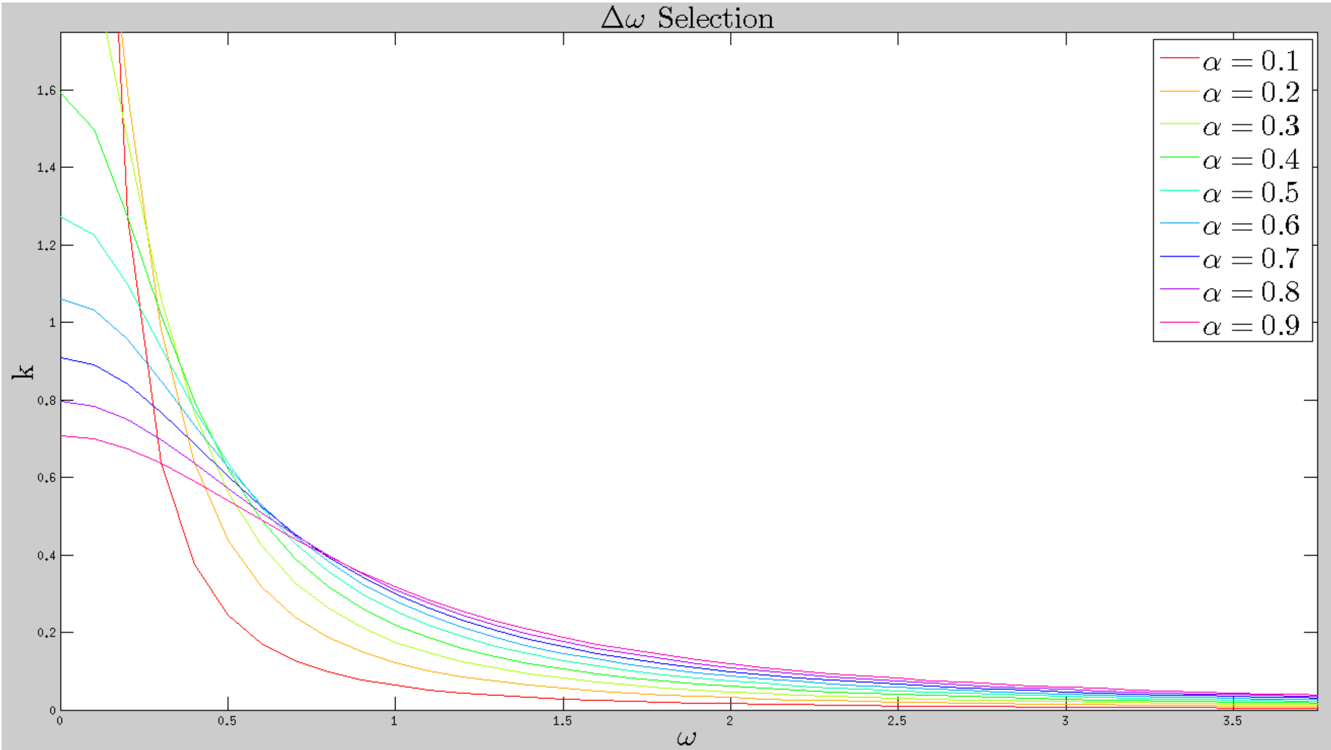

Figure 1: Zeta is chosen as the value ω where the plot of $k(\omega)$ is close enough to zero that the error of not including the rest of these ω values into our estimate is close zero.

II Auto-correlation of Force

The auto-correlation of the force of the large mass, M, is written as

$$< F(0)F(t) >= C(t) - \beta \int_0^t C(t-\tau) \int_0^\tau h(\tau - \tau')C(\tau')d\tau' d\tau$$

and assuming

$$C(t) = \Gamma \alpha e^{-\alpha t}$$

and approximating the integral by the Taylor expansion we get

$$< F(0)F(t) >\approx \Gamma \alpha e^{-\alpha t} - \frac{\beta \Gamma^2}{2M}(\alpha t)^2 + \frac{\beta \Gamma^2}{3M}(\alpha t)^3$$

and thus the approximation that

$$< F(0)F(t) >\approx \Gamma \alpha e^{-\alpha t} \approx C(t) = \beta^{-1}\gamma(t)$$

only holds when

$$\alpha t << 1 \;\; \& \;\; 1 >> \beta(\Gamma \alpha t)^2/M$$

III The Code

In the Matlab code you are required to set the following:

- N: Number of springs
- h: timestep
- T: total simulation time
- *zeta*: choosen as discussed before. *delOmegaSelection.m* will allow you to plot $k(\omega)$ for a given alpha to help in the decision of ζ.
- *N_runs*: number of time to run the simulation.
- *alpha*: value for α
- *objects(1,m)*: mass of the large object, M
- *objects(1,k)*: spring constant of the large object, K

The code executes in the following manner:

- *Find all ω_i, k_i, and m_i values for all N "heat bath" particles*

  FOR 1:N_{runs}

  - *Compute p_i^0 and q_i^0 of each N "heat bath" particle*

    FOR 1:T/h

    - *Compute the force on each "heat bath" particle due to the marked particle*
    - *Update the position of each "heat bath" particle*
    - *Compute the total force due to each "heat bath" particle on the marked particle*
    - *Update the position of the marked particle*

    END

  END

- *Compute averaged force auto-correlation*
- *Plot the force auto-correlation, exponential memory kernel, and summation form of memory kernel*

IV Results

For our test we wanted to recreate simulations where:

$$1 >> \beta\Gamma\alpha t^2/M$$

$$1 > \beta\Gamma\alpha t^2/M$$

$$1 < \beta\Gamma\alpha t^2/M$$

$$1 << \beta\Gamma\alpha t^2/M$$

to do so we set $\beta = 1$ and $\Gamma = 1$ to eliminate their effects. We chose $\alpha = 0.8$ and varied only M, the mass of the marked particle. For our first simulation, (*Fig. 2*), we choose $M = 1000$ which then produced a simulation where $1 >> \beta\Gamma\alpha t^2/M = 0.08$. We noticed that for this simulation the autocorrelation of the force on the marked particle matched the memory kernel perfectly as expected.

For our next simulation, (*Fig. 3*), we choose $M = 100$ which then produced a simulation where $1 > \beta\Gamma\alpha t^2/M = 0.8$. We noticed that for this simulation the autocorrelation of the force on the marked particle matched the memory kernel only when $\alpha t < 1$ where at any αt around 1 or greater then one the autocorrelation of the force was quite.

In our third simulation, (*Fig. 4*), we choose $m = 10$ which then produced a simulation where $1 < \beta\Gamma\alpha t^2/M = 8$. We noticed that for this simulation the autocorrelation of the force on the marked particle matched the memory kernel only when $\alpha t < 1$ where at any αt around 1 or greater then one the autocorrelation of the force was quite different.

In our fourth simulation, (*Fig. 5*), we choose $m = 1$ which then produced a simulation where $1 << \beta\Gamma\alpha t^2/M = 80$. We noticed that for this simulation the autocorrelation of the force on the marked particle matched the memory kernel only when $\alpha t << 1$. We also noticed a single oscillation before the force autocorrelation converged to 0 not seen in the previous simulations.

In our final two simulations, (*Fig. 6, 7*), we choose $M = 0.1$ and $M = 0.01$ which then produced a simulation where $1 << \beta\Gamma\alpha t^2/M = 800$ and $1 << \beta\Gamma\alpha t^2/M = 8000$, respectively. We noticed that for this simulation the autocorrelation of the force on the marked particle matched the memory kernel only when $\alpha t << 1$ as with the fourth simulation. We also found that as the mass, M, grew increasingly smaller we saw an increasing number of oscillations present in force autocorrelation before eventually decaying to 0.

Overall our results were very much inline with out expectations. As we varied the magnitude of $\beta\Gamma\alpha t^2/M$ we saw changing behaviors in the force autocorrelation of the marked particle as expected. Thus, we have conclusively shown our mathematical results hold true in a real experimental system.

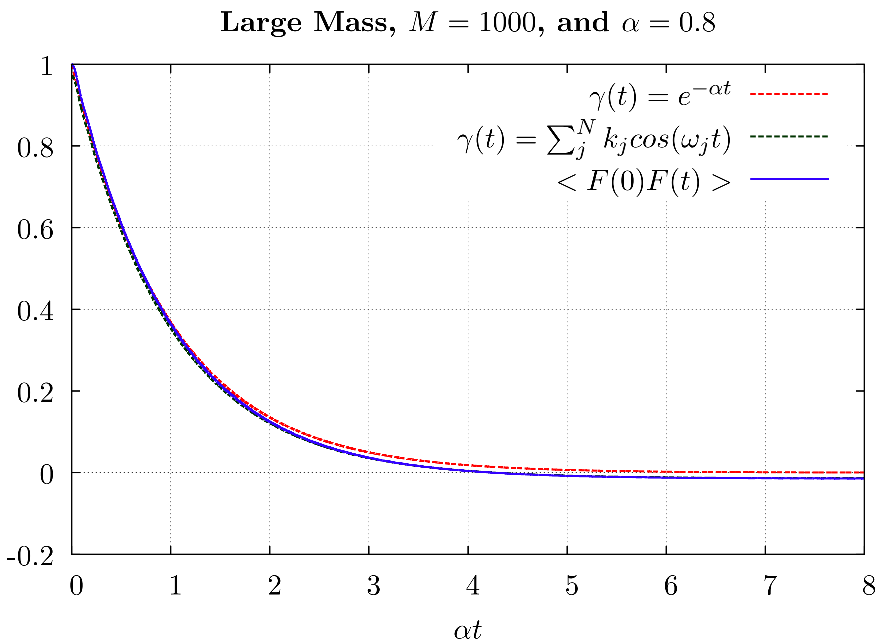

Figure 2: Simulation where $1 >> \beta\Gamma\alpha t^2/M = 0.08$. We can clearly see that the autocorrelation of the force on the marked particle matched that of our memory kernel.

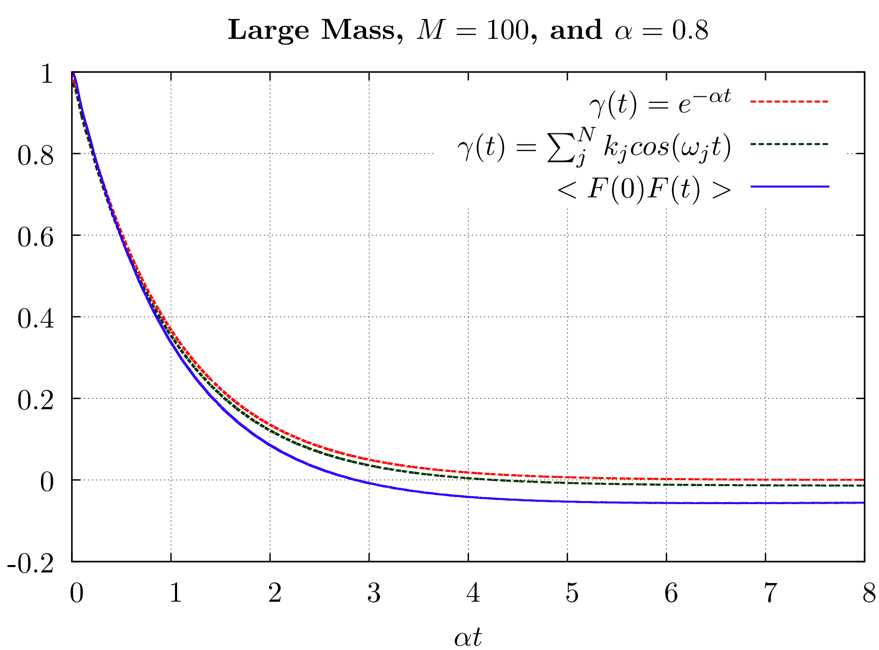

Figure 3: Simulation where $1 > \beta\Gamma\alpha t^2/M = 0.8$. We notice that the autocorrelation of the force on the marked particle begins to diverge from our memory kernel when $\alpha > t$

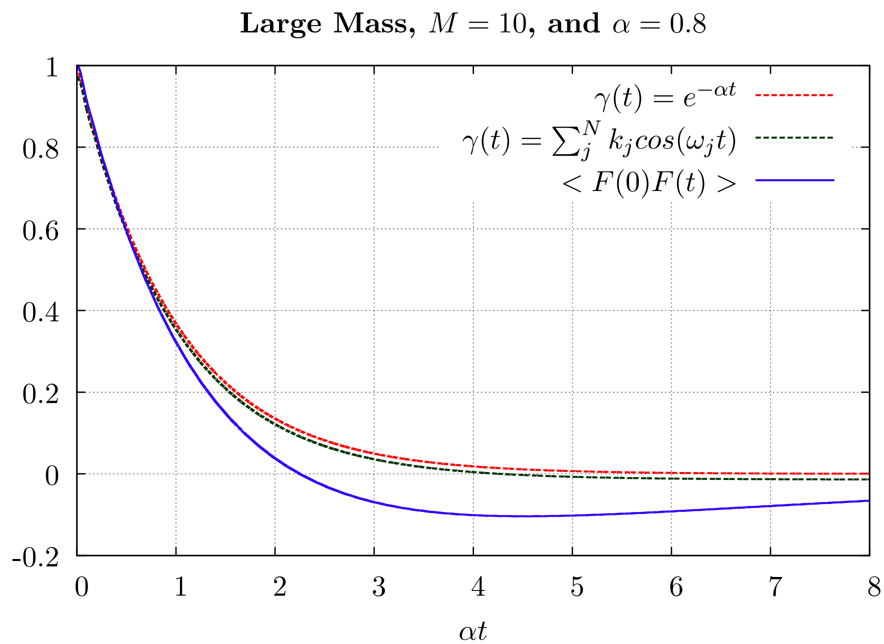

Figure 4: Simulation where $1 < \beta\Gamma\alpha t^2/M = 8$. We notice that the autocorrelation of the force on the marked particle begins to diverge even further from our memory kernel when $\alpha > t$

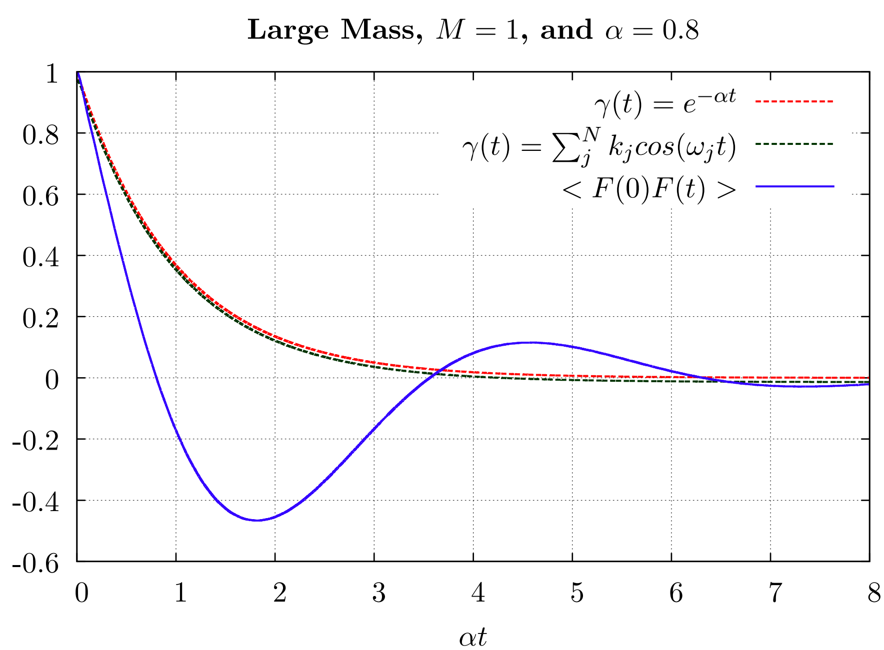

Figure 5: Simulation where $1 << \beta\Gamma\alpha t^2/M = 80$. We notice that the autocorrelation of the force on the marked particle diverges significantly from our memory kernel when $\alpha >> t$ and an single oscillation is present.

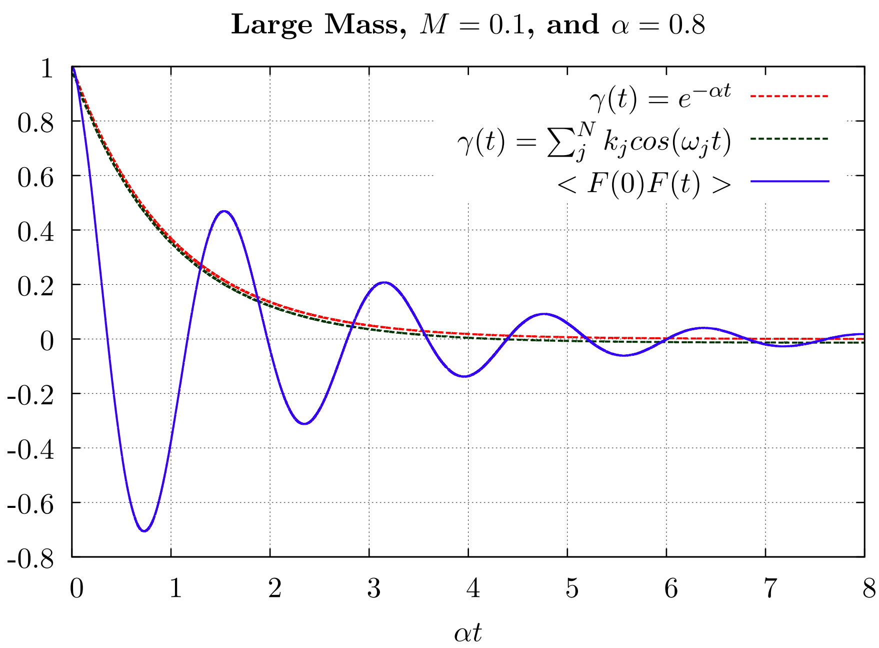

Figure 6: Simulation where $1 << \beta\Gamma\alpha t^2/M = 800$. We notice that the autocorrelation of the force on the marked particle diverges significantly from our memory kernel when $\alpha >> t$ and an multiple oscillations are present.

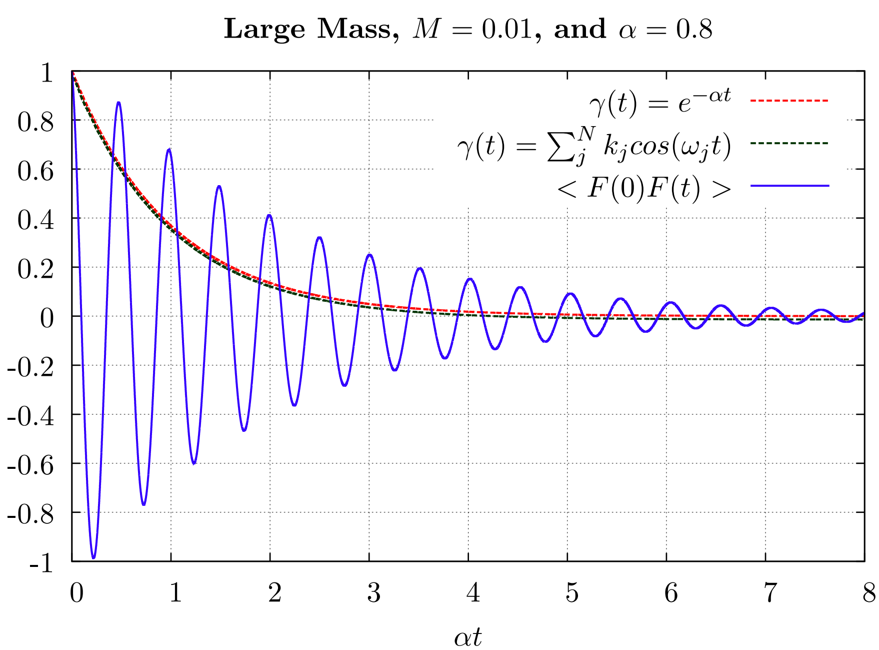

Figure 7: Simulation where $1 << \beta\Gamma\alpha t^2/M = 8000$. We notice that the autocorrelation of the force on the marked particle diverges significantly from our memory kernel when $\alpha >> t$ and even more oscillations are present.