to coarse graining the interactions, and the relation between dynamical algorithms and Monte Carlo methods, to name but two. We discuss these at the end of the chapter.

12.2 Langevin and Brownian dynamics

The Langevin equation is a stochastic differential equation, describing the Brownian motion of particles in a liquid, as well as a number of other physical systems (Chandrasekhar, 1943; Snook, 2007; Coffey and Kalmykov, 2012). Formally, it may be derived by applying projection operator methods to the equations of motion for the phase space distribution function (Zwanzig, 1960; 1961a,b) or the dynamical variables themselves (Mori, 1965a,b); an elegant unified treatment has been presented by Nordholm and Zwanzig (1975). The essential physical idea behind the derivation is *time scale separation*: the variables that are retained are assumed to vary much more slowly than those that are represented by stochastic terms. More details may be found elsewhere (Berne and Pecora, 1976; McQuarrie, 1976; Hansen and McDonald, 2013).

Our starting point is the Langevin equation in the following form:

$$\dot{\mathbf{r}} = \mathbf{v} = \mathbf{p}/m, \quad \dot{\mathbf{p}} = \mathbf{f} - \xi\mathbf{v} + \sigma\dot{\mathbf{w}} = \mathbf{f} - \gamma\mathbf{p} + \sigma\dot{\mathbf{w}}. \tag{12.1}$$

As usual, each vector represents the complete set of N particles. The last three terms are, respectively, the effects of the systematic forces of interaction between the particles, the frictional forces on them due to the solvent, and the so-called random forces, which are represented by the time derivative of a ($3N$-dimensional) Wiener process $\mathbf{w}$. Here the friction coefficient ξ (or equivalently the damping constant γ) is related by the equation

$$\xi = m\gamma = \frac{k_{\mathrm{B}}T}{D}$$

to the diffusion coefficient D of the particles *in the absence of any interactions*, that is, if we set $\mathbf{f} = 0$. The coefficient σ governs the strength of the random forces. It is related to ξ through the *fluctuation–dissipation theorem*

$$\sigma = \sqrt{2\xi k_{\mathrm{B}}T} = \sqrt{2\gamma m k_{\mathrm{B}}T}.$$

Given this equation, and the properties of $\mathbf{w}$ discussed later, it can be shown that eqn (12.1) generates a trajectory that samples states from the canonical ensemble at temperature T. The key relations between ξ, σ, and D essentially go back to Einstein's analysis of Brownian motion; details can be found elsewhere (Chandrasekhar, 1943; Kubo, 1966; Snook, 2007; Marconi et al., 2008; Coffey and Kalmykov, 2012). Simulation methods which use eqn (12.1), as well as the version without inertia which will be given shortly, are generally referred to as Brownian dynamics (BD) techniques, or sometimes as Langevin dynamics.

We should spend a moment discussing the physics, and the mathematics, implicit in eqn (12.1), especially the random force term. Each of the $3N$ components $w_{i\alpha}$ ($i = 1 \ldots N$, $\alpha = x, y, z$), is assumed to be independent. The definition of a Wiener process $w_{i\alpha}$ is that its change over a differentially small time interval dt is a random variable, normally distributed, with variance equal to dt. In other words, we may write

$$\mathrm{d}\mathbf{w} = \mathbf{w}(t + \mathrm{d}t) - \mathbf{w}(t) = \sqrt{\mathrm{d}t}\,\mathbf{G} \tag{12.2}$$

where each component of $\mathbf{G}$, $G_{i\alpha}$, is an independent Gaussian random variable, with zero mean $\langle G_{i\alpha}\rangle = 0$ and unit variance, $\langle G_{i\alpha}G_{j\beta}\rangle = \delta_{ij}\delta_{\alpha\beta}$. So, $\dot{\mathbf{w}}$ is not properly defined, since d$\mathbf{w}$ is not proportional to dt in the limit $\mathrm{d}t \to 0$. The $\sqrt{\mathrm{d}t}$-dependence means that the momentum part of eqn (12.1) should really be written

$$\mathrm{d}\mathbf{p} = \mathbf{f}\,\mathrm{d}t - \gamma\mathbf{p}\,\mathrm{d}t + \sigma\,\mathrm{d}\mathbf{w}$$

and this form, together with eqn (12.2), translates straightforwardly into a numerical integration algorithm with a nonvanishing timestep δt, as we shall see. Before discarding $\dot{\mathbf{w}}$, however, we note that its time correlation function is proportional to a Dirac delta function $\langle \dot{w}_{i\alpha}(0)\dot{w}_{i\alpha}(t)\rangle \propto \delta(t)$. This is consistent with the picture of rapid, random, buffeting of a particle by the surrounding solvent. It is also the defining characteristic of 'white noise', which is a term sometimes used to describe the random force term.

We should note two more important features of the stochastic term: it is *uncorrelated* with any of the dynamical variables (positions and velocities) at earlier times, and it is an *additive* term, which avoids some complications of stochastic calculus. This simplifies the construction of a numerical algorithm. The earliest approach (Ermak and Buckholz, 1980) had the disadvantage of not reducing to a stable molecular dynamics method in the limit of low friction $\xi \to 0$. Later suggestions (Allen, 1980; 1982; van Gunsteren and Berendsen, 1982; 1988) were designed to give Verlet-equivalent algorithms in this limit, and the particular suggestion of Brünger et al. (1984) has been widely adopted.

The fact that the stochastic terms are independent of the coordinates and momenta means that the equations in the absence of forces are exactly soluble, and this allows the construction of a symplectic integration algorithm, using an approach similar to the one that yields velocity Verlet (see Section 3.2.2). An operator-splitting method may be devised in various different ways (see e.g. Cotter and Reich, 2006; Bussi and Parrinello, 2007; Melchionna, 2007). These have been analysed in some detail (Leimkuhler and Matthews, 2013a,b). The optimal method, which they call 'BAOAB', involves inserting, into the middle of the usual 'kick–drift–kick' sequence, the exact solution of the force-free momentum equation

$$\mathrm{d}\mathbf{p} = -\gamma\mathbf{p}\,\mathrm{d}t + \sqrt{2\gamma m k_{\mathrm{B}}T}\,\mathrm{d}\mathbf{w}$$

which is

$$\mathbf{p}(t + \delta t) = \exp(-\gamma\delta t)\mathbf{p}(t) + \sqrt{1 - \exp(-2\gamma\delta t)}\sqrt{m k_{\mathrm{B}}T}\mathbf{G}.$$

Code 12.1 Brownian dynamics program

This file is provided online. The BD programme `bd_nvt_lj.f90`, combined with the standard Lennard-Jones module `md_lj_module.f90` (Code 3.4) for the forces, and the utility modules of Appendix A, carries out Langevin equation dynamics using the BAOAB algorithm of Leimkuhler and Matthews (2013a).

```
! bd_nvt_lj.f90
! Brownian dynamics, NVT ensemble
PROGRAM bd_nvt_lj
```

This gives the following algorithm

$$
\begin{aligned}
\mathbf{p}(t+\tfrac{1}{2}\delta t) &= \mathbf{p}(t) + \tfrac{1}{2}\delta t\mathbf{f}(t) && \text{(12.3a)}\\
\mathbf{r}(t+\tfrac{1}{2}\delta t) &= \mathbf{r}(t) + \tfrac{1}{2}\delta t\mathbf{p}(t+\tfrac{1}{2}\delta t)/m && \text{(12.3b)}\\
\mathbf{p}'(t+\tfrac{1}{2}\delta t) &= \exp(-\gamma\delta t)\mathbf{p}(t+\tfrac{1}{2}\delta t) + \sqrt{1-\exp(-2\gamma\delta t)}\sqrt{mk_{\mathrm{B}}T}\mathbf{G} && \text{(12.3c)}\\
\mathbf{r}(t+\delta t) &= \mathbf{r}(t+\tfrac{1}{2}\delta t) + \tfrac{1}{2}\delta t\mathbf{p}'(t+\tfrac{1}{2}\delta t)/m && \text{(12.3d)}\\
\mathbf{p}(t+\delta t) &= \mathbf{p}'(t+\tfrac{1}{2}\delta t) + \tfrac{1}{2}\delta t\mathbf{f}(t+\delta t). && \text{(12.3e)}
\end{aligned}
$$

Here, eqns (12.3a) and (12.3e) are the (completely standard) half-step kicks. Eqns (12.3b) and (12.3d) are the usual drift equations, but for half a step at a time. The frictional and random force terms together appear in the middle step (12.3c). Leimkuhler and Matthews have shown that this algorithm performs well at both high and low values of the friction, using test systems such as a one-dimensional oscillator, Lennard-Jones and Morse potential atomic clusters, and alanine dipeptide (solvated and unsolvated). An example BD program is given in Code 12.1.

At high friction, the relaxation of the momenta can be assumed to occur instantaneously. Setting $\dot{\mathbf{p}} = 0$ in eqn (12.1) and substituting into the equation for $\dot{\mathbf{r}}$ gives the Brownian dynamics equation of motion (i.e. the Langevin equation without inertia)

$$\dot{\mathbf{r}} = \xi^{-1}\left(\mathbf{f} + \sigma\dot{\mathbf{w}}\right) = \frac{D}{k_{\mathrm{B}}T}\left(\mathbf{f} + \sigma\dot{\mathbf{w}}\right). \tag{12.4}$$

A simple algorithm for this (Ermak and Yeh, 1974; Ermak, 1975) is

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{D}{k_{\mathrm{B}}T}\mathbf{f}(t)\delta t + \sqrt{2D\delta t}\,\mathbf{G}. \tag{12.5}$$

Again, the diffusion coefficient enters as a parameter of the method, and determines the variance of the last term, the random Gaussian displacements. In the absence of forces, we would see displacements $\delta r_{i\alpha} = r_{i\alpha}(t+\delta t) - r_{i\alpha}(t)$ satisfying $\langle \delta r_{i\alpha}^2 \rangle = 2D\delta t$ as expected. This algorithm can also be interpreted as a Monte Carlo method of the kind discussed in Section 9.3, and we return to this in Section 12.3. The formalism may easily be extended from atomic systems to include rigid and non-rigid molecules, and the incorporation of constraints is straightforward (van Gunsteren and Berendsen, 1982) although the usual care should be taken in their application (van Gunsteren, 1980; van Gunsteren and Karplus, 1982).

The previous equations all ignore memory effects in the random force as well as indirect interactions between the atoms, mediated by the solvent. In principle, the inclusion of a specified memory function into the Langevin equation is straightforward (Ciccotti et al., 1976a; Doll and Dion, 1976; Adelman, 1979; Ermak and Buckholz, 1980; Ciccotti and Ryckaert, 1980). Essentially, this corresponds to the inclusion of additional derivatives of the momentum in the equations of motion, or to the extension of the time correlation function of the random force from a delta function to, for example, a decaying exponential.

The simplest effect of the surrounding solvent is to replace the bare interaction between solute particles by a potential of mean force. We shall discuss this in Section 12.7. The second effect, neglected in the simple Langevin and Brownian dynamics equations, is the effect that solvent flow, induced by one molecule, has on the surrounding molecules. If this hydrodynamic effect is not tackled directly (as in Sections 12.4–12.6), it can be introduced approximately into the Brownian dynamics algorithm via a configuration-dependent diffusion coefficient. The way this appears in the equation of motion depends on the convention adopted for stochastic differentials; however, the integration algorithm is unambiguously written (Ermak and McCammon, 1978)

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{\mathbf{D}(t)}{k_\mathrm{B}T}\cdot\mathbf{f}(t)\delta t + \boldsymbol{\nabla}\cdot\mathbf{D}(t)\delta t + \mathbf{R}. \tag{12.6}$$

Here, as usual, the vectors contain $3N$ components; $\mathbf{D}$ is a $3N \times 3N$ diffusion tensor or matrix, whose components depend on molecular positions. The random part of the displacement, $\mathbf{R}$, is selected from the $3N$-variate Gaussian distribution with zero means and covariance matrix

$$\big\langle \mathbf{R}\mathbf{R}\big\rangle = 2\mathbf{D}\delta t. \tag{12.7}$$

As a consequence, the components of $\mathbf{R}$ are *correlated* with each other. Sampling these random variables is a comparatively time-consuming exercise, depending on some expensive manipulations of the matrix $\mathbf{D}$. Two forms of $\mathbf{D}$ are commonly adopted. The simplest, suggested by the equations of macroscopic hydrodynamics, is the Oseen tensor. Writing $\mathbf{D}$ as an $N \times N$ set of 3×3 matrices $\mathbf{D}_{ij}$ for each pair of molecules,

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1N} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{N1} & \mathbf{D}_{N2} & \cdots & \mathbf{D}_{NN} \end{pmatrix}$$

this takes the form

$$\mathbf{D}_{ij} = \begin{cases} \dfrac{k_\mathrm{B}T}{6\pi\eta a}\mathbf{1} & i=j \\ \dfrac{k_\mathrm{B}T}{8\pi\eta r_{ij}}\Big(\mathbf{1} + \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}\Big) & i\neq j \end{cases} \tag{12.8}$$

where η is the viscosity, a an estimate of the hydrodynamic radius (not diameter!), and $\mathbf{r}_{ij}$ the vector between the molecules, with $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$ as usual. This tensor has the property that $\boldsymbol{\nabla} \cdot \mathbf{D} = 0$, so this term may be dropped from eqn (12.6). The standard approach to generating the random displacements is to factorize $\mathbf{D}$ by the Cholesky square root method (Press et al., 2007), that is, determine the (unique) lower triangular real matrix $\mathbf{L}$ such that $\mathbf{D} = \mathbf{L} \cdot \mathbf{L}^{\mathrm{T}}$ (where $\mathbf{L}^{\mathrm{T}}$ is the transpose of $\mathbf{L}$). Then, given *independently* sampled normal (unit-variance) variables $\mathbf{G}$,

$$\mathbf{R} = \sqrt{2\Delta t}\,\mathbf{L} \cdot \mathbf{G} \tag{12.9}$$

will be the desired set of *correlated* Gaussian displacements (Ermak and McCammon, 1978) (see also Appendix E). However, the decomposition is only valid for positive definite matrices, and the Oseen tensor does not always fulfil this condition.

For this reason, the Oseen tensor is commonly replaced by the Rotne–Prager–Yamakawa tensor (Rotne and Prager, 1969; Yamakawa, 1970), which has the identical form for $i = j$, but which for different particles $i \neq j$ is

$$\mathbf{D}_{ij} = \begin{cases} \left(\dfrac{k_{\mathrm{B}}T}{8\pi\eta r_{ij}}\right)\left[\left(\mathbf{1} + \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}\right) + \dfrac{2a^2}{r_{ij}^2}\left(\tfrac{1}{3}\mathbf{1} - \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}\right)\right] & \text{for } r_{ij} \geq 2a, \\ \left(\dfrac{k_{\mathrm{B}}T}{6\pi\eta a}\right)\left[\left(1 - \dfrac{9}{32}\dfrac{r_{ij}}{a}\right)\mathbf{1} + \dfrac{3}{32}\dfrac{r_{ij}}{a}\hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}\right] & \text{for } r_{ij} < 2a. \end{cases} \tag{12.10}$$

This also has the property of zero divergence, and is positive definite for all r_{ij}. We should bear in mind that all these tensors are only leading approximations (Felderhof, 1977; Schmitz and Felderhof, 1982), and what is more they assume pairwise additivity in what is really a many-body problem in hydrodynamics (Mazur, 1982; Mazur and van Saarloos, 1982; van Saarloos and Mazur, 1983).

The solution of these equations of motion may be speeded up by a variety of techniques. The Cholesky method has computational cost $O(N^3)$ which rapidly becomes expensive for large N. Fixman (1986) has proposed a method for improving this to $O(N^{2.25})$, using an expansion in Chebyshev polynomials. Beenakker (1986) has suggested handling the long-range hydrodynamic interactions by Ewald sum; there are some subtleties associated with this (Smith, 1987; Smith et al., 1987). Banchio and Brady (2003) apply a fast Fourier transform method to compute many-body long-range hydrodynamic interactions, and divide the Brownian forces into near-field and far-field components, resulting in a method that is $O(N^{1.5} \log N)$ for sufficiently large N. Jain et al. (2012) have optimized both the Chebyshev and Ewald elements of the calculation, for semidilute polymer solutions, giving an overall cost $O(N^{1.8})$. More recently, a method called the truncated expansion approximation (Geyer and Winter, 2009) was proposed to calculate the correlated random displacements more rapidly, and a technique using Krylov subspaces (Ando et al., 2012) seems very promising.

12.3 Brownian dynamics, molecular dynamics, and Monte Carlo

The BD method may be related to MD, and MC, techniques (Rossky et al., 1978; Horowitz, 1991; Akhmatskaya et al., 2009; Allen and Quigley, 2013). Starting with eqn (12.5), and making the change of variables

$$D = \frac{k_{\mathrm{B}}T}{2m}\delta t$$