1 Numerical Integrators

We are now at the crucial step of the MD code, the integration of the equation of motion, e.g. the build up of the system trajectory in phase-space. We need to find algorithms that preserves time invariance and phase-space compressibility. These conditions are crucial to generate algorithms that do not present significant drift in the total energy with time.

1.1 Verlet

The most common integrator (Verlet integrator) originates from the addition between the future and past Taylor expansion of the trajectory (that requires information of position, velocity and acceleration at time t)

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t)$$

and

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \Delta t \mathbf{v}_i(t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t)$$

Summing up the two equations to eliminate the velocity gives

$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \Delta t^2 \mathbf{a}_i(t)$$

which can be rewritten as

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \Delta t^2 \frac{\mathbf{F}_i(t)}{m} \tag{1}$$

Eq. 41 provides the trajectory of the particles building only on information of the previous position. The velocity, if needed, can be calculated as

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t-\Delta t)}{2\Delta t}$$

1.1.1 Time reversal of the Verlet algorithm

Let's write the trajectory according to Verlet at time $t+\Delta t$ and $t+2\Delta t$

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \Delta t^2 \frac{\mathbf{F}_i(t)}{m} \tag{2}$$

$$\mathbf{r}_i(t+2\Delta t) = 2\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t) + \Delta t^2 \frac{\mathbf{F}_i(t+\Delta t)}{m} \tag{3}$$

If we are in $\mathbf{r}_i(t+\Delta t)$ and decide to go back in time (by $-\Delta t$) we should write [$(t \to t+\Delta t)$ and $\Delta t \to -\Delta t$)]

$$\mathbf{r}_i^{back}(t+\Delta t-\Delta t) = 2\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t+\Delta t-(-\Delta t)) + \Delta t^2 \frac{\mathbf{F}_i(t+\Delta t)}{m}$$

Substituting the force at $t+\Delta t$ from Eq. 3 one gets

$$\mathbf{r}_i^{back}(t) = 2\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t+2\Delta t) + \mathbf{r}_i(t+2\Delta t) - 2\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t) = \mathbf{r}_i(t)$$

proving the time reversal of the Verlet algorithm.

1.2 Leap-Frog

The Leap-Frog algorithm is equivalent to Verlet (i.e. it provides the same trajectory as Verlet). It only differs in the definition of the velocities.

We start from the numerical definition of derivatives for $\mathbf{v}$ and $\mathbf{a}$

$$\mathbf{v}_i(t+\Delta t/2) = \frac{\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t)}{\Delta t}$$

and

$$\mathbf{a}_i(t) = \frac{\mathbf{v}_i(t+\Delta t/2) - \mathbf{v}_i(t-\Delta t/2)}{\Delta t}$$

These two equations define, once written in the form

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \mathbf{v}(t+\Delta t/2)\Delta t \tag{4}$$

$$\mathbf{v}_i(t+\Delta t/2) = \mathbf{v}_i(t-\Delta t/2) + \Delta t\mathbf{a}_i(t) \tag{5}$$

the Leap-Frog algorithm.

To verify that the Leap-Frog is equivalent to Verlet, let's consider

$$\mathbf{v}_i(t-\Delta t/2) = \frac{\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t)}{\Delta t}$$

Substituting Eq. 5 in Eq. 4 and the previous expression for $\mathbf{v}_i(t-\Delta t/2)$ one gets

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\mathbf{v}_i(t-\Delta t/2) + \Delta t^2\mathbf{a}_i(t) = \mathbf{r}_i(t) + \Delta t(\frac{\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t)}{\Delta t}) + \Delta t^2\mathbf{a}_i(t)$$

$$= 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \Delta t^2\mathbf{a}_i(t)$$

that coincides with the Verlet trajectory.

1.3 Velocity Verlet

A more accurate related algorithm is based on a equal treating of position and momenta. Starting again from the Taylor expansions of $\mathbf{r}_i(t+\Delta t)$ around $\mathbf{r}_i(t)$ one get

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t) \tag{6}$$

Similarly, expanding $\mathbf{r}_i(t)$ this time around $\mathbf{r}_i(t+\Delta t)$ (e.g. back in time) one obtains

$$\mathbf{r}_i(t) = \mathbf{r}_i(t+\Delta t) - \Delta t \mathbf{v}_i(t+\Delta t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t+\Delta t)$$

These two expressions can be combined, eliminating all position informations, in

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2}[\mathbf{a}_i(t) + \mathbf{a}_i(t+\Delta t)] = \mathbf{v}_i(t) + \frac{\Delta t}{2m}[\mathbf{F}_i(t+\Delta t) + \mathbf{F}_i(t)] \tag{7}$$

Eq. 7, together with the Taylor expansion of the positions (Eq. 6) evolves simultaneously both position and velocities. First it evolves positions, then evaluate the forces at the new position, then evolves the velocities. The iteration of this cycles generates the complete trajectory in phase-space.

Both the Verlet and the velocity Verlet algorithms satisfy the two main requests of a stable MD propagation algorithm. Both algorithms indeed satisfy time reversal (e.g. from the position at time $t+\Delta t$ one recovers exactly the original position at time t if a negative Δt is chosen). Both algorithms are also simplectic, e.g. preserve the volume in phase space, as we will discuss later on.

2 Brief background on Lagrangian, Hamiltonian, Compressibility, Symplectic structure

2.1 Equation of motion in the Hamilton representation

The Hamilton equation of motion for $\mathbf{q}$ and $\mathbf{p}$ are

$$\dot{q_\alpha} = \frac{\partial \mathcal{H}}{\partial p_\alpha}$$

and

$$\dot{p_\alpha} = -\frac{\partial \mathcal{H}}{\partial q_\alpha}$$

where $\mathcal{H}(\mathbf{q}, \mathbf{p})$ is the Hamiltonian of the system.

There are two important properties of the Hamilton equation that should be preserved in a numerical description of the dynamics: (i) the conservation of energy and (ii) the incompressibility of the phase space.

2.2 (i) Energy Conservation

Hamilton's equations conserve the total Hamiltonian:

$$\frac{d\mathcal{H}}{dt} = 0$$

Since $\mathcal{H}$ is the total energy, this equation is just the law of energy conservation. In order to see that $\mathcal{H}$ is conserved, we simply compute the time derivative via the chain rule in generalized coordinates

$$\frac{d\mathcal{H}}{dt} = \sum_{\alpha} \left[\frac{\partial \mathcal{H}}{\partial q_\alpha} \frac{\partial \mathcal{H}}{\partial p_\alpha} - \frac{\partial \mathcal{H}}{\partial p_\alpha} \frac{\partial \mathcal{H}}{\partial q_\alpha} \right]$$

The conservation of energy defines a $3N-1$ dimensional surface in the phase space on which a trajectory must remain. This surface is known as the constant-energy hypersurface or simply the constant-energy surface.

2.3 (ii) Incompressibility of the phase space

Another fundamental property of Hamilton's equations is known as the condition of phase space incompressibility. To understand this condition, consider writing Hamilton's equations directly in terms of the phase space vector as

$$\dot{x} = \eta(x)$$

where $\eta(x)$ is a vector function of the phase space vector x. Since

$$x = (q_1, q_2 ... q_{3N}, p_1, ... p_{3N})$$

it follows that

$$\dot{x} = \left(\frac{\partial \mathcal{H}}{\partial p_1}, ..., \frac{\partial \mathcal{H}}{\partial p_{3N}}, -\frac{\partial \mathcal{H}}{\partial q_1}, ..., -\frac{\partial \mathcal{H}}{\partial q_{3N}} \right)$$

This equation illustrates the fact that the general phase space "velocity" $\dot{x}$ is a function of x, suggesting that motion in phase space can be regarded as a kind of "flow field" as in hydrodynamics, where one has a physical velocity flow field, $v(r)$. Thus, at each point in phase space, there will be a velocity vector $\dot{x}(x)$ equal to $\eta(x)$. In hydrodynamics, the condition for incompressible flow is that there be no sources or sinks in the flow, expressible as $\nabla \cdot \mathbf{v} = 0$. In phase space flow, the analogous condition is $\nabla_x \cdot \dot{x} = 0$, where $\nabla_x = \partial / \partial x$ is the phase space gradient operator. Hamilton's equations of motion guarantee that the incompressibility condition in phase space is satisfied. To see this, consider the compressibility in generalized coordinates

$$\nabla_x \cdot \dot{x} = \sum_{\alpha=1}^{3N} \left[\frac{\partial \dot{q}_\alpha}{\partial q_\alpha} + \frac{\partial \dot{p}_\alpha}{\partial p_\alpha} \right] = \sum_{\alpha=1}^{3N} \left[\frac{\partial}{\partial q_\alpha} \frac{\partial \mathcal{H}}{\partial p_\alpha} - \frac{\partial}{\partial p_\alpha} \frac{\partial \mathcal{H}}{\partial q_\alpha} \right] = 0$$

3 The classical time evolution operator (Tuckerman)

Thus far, we have discussed numerical integration in a somewhat simplistic way, relying on Taylor series expansions to generate update procedures. However, because there are certain formal properties of Hamiltonian systems that should be preserved by numerical integration methods, it is important to develop a formal structure that allows numerical solvers to be generated more rigorously. The framework we seek is based on the classical time evolution operator approach, and we will return to this framework repeatedly throughout the book.

We begin by considering the time evolution of any function $a(x)$ of the phase space vector. If $a(x)$ is evaluated along a trajectory x_t, then in generalized coordinates, the time derivative of $a(x_t)$ is given by the chain rule

$$\frac{da}{dt} = \sum_{i,\alpha} \left(\frac{\partial a}{\partial r_{i,\alpha}} \dot{r}_{i,\alpha} - \frac{\partial a}{\partial p_{i,\alpha}} \dot{p}_{i,\alpha} \right)$$

Inserting the classical law of evolution provided by the Hamilton equations

$$\frac{dr_{i,\alpha}}{dt} = \frac{\partial \mathcal{H}}{\partial p_{i,\alpha}} \qquad \frac{dp_{i,\alpha}}{dt} = -\frac{\partial \mathcal{H}}{\partial r_{i,\alpha}}$$

one obtains

$$\frac{da}{dt} = \sum_{i,\alpha} \left(\frac{\partial a}{\partial r_{i,\alpha}} \frac{\partial \mathcal{H}}{\partial p_{i,\alpha}} - \frac{\partial a}{\partial p_{i,\alpha}} \frac{\partial \mathcal{H}}{\partial r_{i,\alpha}} \right) = \{a, \mathcal{H}\} = -\{\mathcal{H}, a\}$$

By defining a Liouville operatore $\mathcal{L}$ as

$$i\mathcal{L}a = \{a, \mathcal{H}\}$$

(where the imaginary i is included to stress the analogy with the corresponding quantum propagator) we can write the symbolic evolution of a as

$$\frac{da}{dt} = i\mathcal{L}a$$

and the formal solution

$$a(x_t) = e^{i\mathcal{L}t} a(x_0) \tag{8}$$

If we identify $a(x_t)$ with x_t then

$$x_t = e^{i\mathcal{L}t} x_0 \tag{9}$$

we see that $e^{i\mathcal{L}t}$ is the operator that propagates the trajectory in phase-space.

Although elegant in its compactness, Eq. 9 amounts to little more than a formal device since we cannot evaluate the action of the operator $e^{i\mathcal{L}t}$ on x_0 exactly. If we could, then any and every problem in classical mechanics could be solved exactly analytically and we

would not be in the business of developing numerical methods in the first place! What Eq. 9 does do is it provides us with a very useful starting point for developing approximate solutions to Hamilton's equations. The Liouville operator can be written as a sum of two contributions

$$i\mathcal{L} = i\mathcal{L}_1 + i\mathcal{L}_2$$

$$i\mathcal{L}_1 = \sum_{i,\alpha} \frac{\partial \mathcal{H}}{\partial p_{i,\alpha}} \frac{\partial ...}{\partial r_{i,\alpha}}$$

$$i\mathcal{L}_2 = -\sum_{i,\alpha} \frac{\partial \mathcal{H}}{\partial r_{i,\alpha}} \frac{\partial ....}{\partial p_{i,\alpha}}$$

Unfortunately, $\mathcal{L}_1$ and $\mathcal{L}_2$ are examples of non-commuting operators and the order by which the two operators are applied is important.

3.1 Test of the commuting properties for an harmonic oscillator

In the case of a one dimensional harmonic oscillator,

$$H(x,p) = \frac{p^2}{2m} + U(x)$$

$$iL_1 = \frac{p}{m}\frac{\partial}{\partial x}$$

$$iL_2 = F(x)\frac{\partial}{\partial p}$$

To test for the commuting property, let's apply L_1L_2 and L_2L_1 to a generic function $\phi(x,p)$.

$$(iL_1)(iL_2)\phi(x,p) = \left(\frac{p}{m}\frac{\partial}{\partial x}\right)\left(F(x)\frac{\partial}{\partial p}\right)\phi(x,p)$$

and working from the right

$$(iL_1)(iL_2)\phi(x,p) = \left(\frac{p}{m}\frac{\partial}{\partial x}\right)\left(F(x)\frac{\partial \phi(x,p)}{\partial p}\right) =$$

$$= \frac{p}{m}\left(\frac{dF(x)}{dx}\frac{\partial \phi(x,p)}{\partial p} + F(x)\frac{\partial^2 \phi(x,p)}{\partial x \partial p}\right)$$

and

$$(iL_2)(iL_1)\phi(x,p) = \left(F(x)\frac{\partial}{\partial p}\right)\left(\frac{p}{m}\frac{\partial}{\partial x}\right)\phi(x,p) =$$

$$\left(F(x)\frac{\partial}{\partial p}\right)\left(\frac{p}{m}\frac{\partial\phi(x,p)}{\partial x}\right) =$$
$$\frac{F(x)}{m}\left(\frac{\partial\phi(x,p)}{\partial x} + p\frac{\partial^2\phi(x,p)}{\partial p\partial x}\right)$$

Since $\phi(x,p)$ is a generic function

$$[iL_1, iL_2] = \frac{p}{m}\frac{dF(x)}{dx}\frac{\partial}{\partial p} - \frac{F(x)}{m}\frac{\partial}{\partial x}$$

proving that the two operators do not commute.

3.2 General case: Trotter

Since $\mathcal{L}_1$ and $\mathcal{L}_2$ generally do not commute, the classical propagator $\exp(i\mathcal{L}t) = \exp[(i\mathcal{L}_1 + i\mathcal{L}_2)t]$ cannot be separated into a simple product $\exp(i\mathcal{L}_1 t)\exp(i\mathcal{L}_2 t)$. This is unfortunate because in many instances, the action of the individual operators $\exp(i\mathcal{L}_1 t)$ and $\exp(i\mathcal{L}_2 t)$ on the phase space vector can be evaluated exactly. Thus, it would be useful if the propagator could be expressed in terms of these two factors. In fact, there is a way to do this using an important theorem known as the Trotter theorem (Trotter, 1959). This theorem states that for two operators A and B for which $[A,B] \neq 0$,

$$e^{(A+B)} = \lim_{P\to\infty}\left[e^{B/2P}e^{A/P}e^{B/2P}\right]^P \tag{10}$$

where P is an integer. In fact, Eq. 10 is commonly referred to as the symmetric Trotter theorem or Strang splitting formula (Strang, 1968). The proof of the Trotter theorem is somewhat involved and can be found in the Appendix C of the Tuckerman book for interested readers. Applying the symmetric Trotter theorem to the classical propagator yields

$$e^{i\mathcal{L}t} = e^{i\mathcal{L}_1 t + i\mathcal{L}_2 t} = \lim_{\Delta t\to 0}\left[e^{i\mathcal{L}_2\Delta t/2}e^{i\mathcal{L}_1\Delta t}e^{i\mathcal{L}_2\Delta t/2}\right]^{(t/\Delta t)} \tag{11}$$

Interestingly, we could apply this expression for $t = \Delta t$ (with the constraint of $\Delta t \to 0$) , obtaining

$$e^{i\mathcal{L}\Delta t} = e^{i\mathcal{L}_1 t + i\mathcal{L}_2 t} \approx e^{i\mathcal{L}_2\Delta t/2}e^{i\mathcal{L}_1\Delta t}e^{i\mathcal{L}_2\Delta t/2} \tag{12}$$

The utility of Eq. 12 is that if the contributions $\mathcal{L}_1$ and $\mathcal{L}_2$ to the Liouville operator are chosen such the action of the operators $\exp(\mathcal{L}_1\Delta t)$ and $\exp(\mathcal{L}_2\Delta t)$ can be evaluated analytically, then Eq. 12 can be used as a numerical propagation scheme for a single time step.

Before progressing, let's look how a generic exponential operator $e^{c\frac{\partial}{\partial y}}$, where c is a constant (independent on y) acts on a function $g(y)$.

$$e^{c\frac{\partial}{\partial y}}g(y) = \sum_{k=0}^{\infty} \frac{1}{k!}\left(c\frac{\partial}{\partial y}\right)^k g(y) = \sum_{k=0}^{\infty} \frac{1}{k!} c^k \frac{\partial^k g(y)}{\partial y^k} = g(y+c) \tag{13}$$

where in the last step we have exploited the Taylor expansion of the function $g(y+c)$ around y. Note that, if $g(y)$ is a constant (e.g. is a function of x – e.g. $h(x)$ – which does not dependent on y) only the first term in the sum remains and

$$e^{c\frac{\partial}{\partial y}}h(x) = \sum_{k=0}^{\infty} \frac{1}{k!}\left(c\frac{\partial}{\partial y}\right)^k h(x) = h(x) \tag{14}$$

We are now in a position to apply the Trotter factorisation scheme to the classical evolution of a system. Choosing an Hamiltonian as

$$H = \sum_i \frac{\mathbf{p}_i^2}{2m} + U(\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_N)$$

$$i\mathcal{L}_1 = \sum_{i,\alpha} \frac{p_{i,\alpha}}{m}\frac{\partial ...}{\partial r_{i,\alpha}} = i\mathcal{L}_1 = \sum_{i=1}^{N} \frac{\mathbf{p}_i}{m}\cdot\frac{\partial ...}{\partial \mathbf{r}_i}$$

$$i\mathcal{L}_2 = \sum_{i,\alpha} F_{i,\alpha}\frac{\partial ....}{\partial p_{i,\alpha}} = \sum_{i=1}^{N} \mathbf{F}_i \cdot \frac{\partial ....}{\partial \mathbf{p}_i}$$

so that

$$\begin{bmatrix} \mathbf{r}_i(\Delta t) \\ \mathbf{p}_i(\Delta t) \end{bmatrix} = \exp\left[\frac{\Delta t}{2}\mathbf{F}_i\cdot\frac{\partial}{\partial \mathbf{p}_i}\right]\exp\left[\Delta t\frac{\mathbf{p}_i}{m}\cdot\frac{\partial}{\partial \mathbf{r}_i}\right]\exp\left[\frac{\Delta t}{2}\mathbf{F}_i\cdot\frac{\partial}{\partial \mathbf{p}_i}\right]\begin{bmatrix} \mathbf{r}_i \\ \mathbf{p}_i \end{bmatrix} \tag{15}$$

operating one by one from right to left we obtain

$$\begin{bmatrix} \mathbf{r}_i(\Delta t) \\ \mathbf{p}_i(\Delta t) \end{bmatrix} = \exp\left[\frac{\Delta t}{2}\mathbf{F}_i\cdot\frac{\partial}{\partial \mathbf{p}_i}\right]\exp\left[\Delta t\frac{\mathbf{p}_i}{m}\cdot\frac{\partial}{\partial \mathbf{r}_i}\right]\begin{bmatrix} \mathbf{r}_i \\ \mathbf{p}_i + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N) \end{bmatrix} = \tag{16}$$

$$\exp\left[\frac{\Delta t}{2}\mathbf{F}_i\cdot\frac{\partial}{\partial \mathbf{p}_i}\right]\begin{bmatrix} \mathbf{r}_i + \Delta t\frac{\mathbf{p_i}}{m} \\ \mathbf{p}_i + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N + \Delta t\frac{\mathbf{p_i}}{m}) \end{bmatrix} = \tag{17}$$

$$\begin{bmatrix} \mathbf{r}_i + \frac{\Delta t}{m}(\mathbf{p_i} + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N)) \\ \mathbf{p}_i + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N) + \frac{\Delta t}{2}\mathbf{F}_i\{\mathbf{r}^N + \frac{\Delta t}{m}(\mathbf{p}_i + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N)\} \end{bmatrix} \tag{18}$$

The result is thus (by going from $\mathbf{p}$ to $\mathbf{v}$, and stressing the previous time as $t = 0$)

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0) + \Delta t\mathbf{v}_i(0) + \frac{\Delta t^2}{2m}\mathbf{F}_i(\mathbf{r}^N(0))$$

and

$$\mathbf{v}_i(\Delta t) = \mathbf{v}_i(0) + \frac{\Delta t}{2m}\left[\mathbf{F}_i(\mathbf{r}^N(0)) + \mathbf{F}_i(\mathbf{r}^N(\Delta t))\right]$$

which coincide with the velocity Verlet algorithm. It is also interesting to observe that the factorisation implied by the Trotter approach matches with a sequential update of positions and velocities. Indeed, we can see Eq. 15 as the sequential application of

$$\mathbf{p}(\Delta t/2) = \mathbf{p}(\mathbf{0}) + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N(0))$$

$$\mathbf{r}(\Delta t) = \mathbf{r}(0) + \frac{\Delta t}{m}\mathbf{p}(\Delta t/2)$$

and

$$\mathbf{p}(\Delta t) = \mathbf{p}(\Delta t/2) + \frac{\Delta t}{2}\mathbf{F}_i(\mathbf{r}^N(\Delta t))$$

that very nicely resemble lines of a computer code. The third line involves a call to some function or subroutine that updates the force from the new positions generated in the second line. When written this way, the specific instructions are: i) perform a momentum translation; ii) follow this by a position translation; iii) recalculate the force using the new position; iv) use the new force to perform a momentum translation. The fact that instructions in computer code can be written directly from the operator factorization scheme, bypassing the lengthy algebra needed to derive explicit finite-difference equations, is an immensely powerful technique that we term the direct translation method (Martyna et al., 1996). Because direct translation is possible, we can simply let a factorization of the classical propagator denote a particular integration algorithm; we will employ the direct translation technique in many of our subsequent discussions of numerical solvers.

4 Multiple time scales (from Tuckerman)

One of the most ubiquitous aspects of complex systems in classical mechanics is the presence of forces that generate motion with different time scales. Examples include long biological macromolecules such as proteins as well as other types of polymers. In fact, virtually any chemical system will span a wide range of time scales from very fast bond and bend vibrations to global conformational changes in macromolecules or slow diffusion/transport molecular liquids, to illustrate just a few cases.

4.1 The FENE (finitely extensible nonlinear elastic) potential

One possible way to simulated a bonded pair is to include a interaction potential between the two bonded atoms defined (for $r < R_0$)

$$V^{FENE}(r) = -\frac{1}{2}kR_0^2 \ln\left[1 - \left(\frac{r}{R_0}\right)^2\right]$$

where R_0 is the maximum extension of the bond and k plays the role of elastic constant. For $r > R_0$ $V^{FENE} = 0$. For small r the standard repulsion confined the bond. Often the FENE potential is used in combination with the cut and shifted LJ potential, cut at the minimum ($r = 2^{1/6}\sigma$) and shifted by ϵ.

On the time scale over which the fast forces vary naturally, the slow forces change very little. In the simple velocity Verlet scheme, one time step δt is employed whose magnitude is limited by the fast forces, yet all force components must be computed at each step, including those that change very little over a time δt. Ideally, it would be advantageous to develop a numerical solver capable of exploiting this separation of time scales for a gain in computational efficiency. Such an integrator should allow the slow forces to be recomputed less frequently than the fast forces, thereby saving the computational overhead lost by updating the slow forces every step. The Liouville operator formalism allows this to be done in a rigorous manner, leading to a symplectic, time-reversible multiple time-scale solver. We will show how the algorithm is developed using, once again, the example of a single particle in one dimension. Suppose the particle is subject to a force, $F(x)$, that has two components, $F_{fast}(x)$ and $F_{slow}(x)$. The equations of motion are

$$\dot{x} = \frac{p}{m}$$
$$\dot{p} = F_{fast}(x) + F_{slow}(x)$$

The associated Loiville operator is

$$i\mathcal{L} = \frac{p}{m}\frac{\partial}{\partial x} + [F_{fast}(x) + F_{slow}(x)]\frac{\partial}{\partial p}$$

that we can split as

$$i\mathcal{L}_{fast} = \frac{p}{m}\frac{\partial}{\partial x} + F_{fast}(x)\frac{\partial}{\partial p}$$

and

$$i\mathcal{L}_{slow} = F_{slop}(x)\frac{\partial}{\partial p}$$

and applying the Trotter scheme

$$e^{i\mathcal{L}\Delta t} = e^{i\mathcal{L}_{slow}\Delta t/2}e^{i\mathcal{L}_{fast}\Delta t}e^{i\mathcal{L}_{slow}\Delta t/2} \tag{19}$$

This factorization leads to the reference system propagator algorithm or RESPA for short (Tuckerman et al., 1992). The idea behind the RESPA algorithm is that the step Δt appearing in Eq. 19 is chosen according to the time scale of the slow forces. There are two ways to achieve this: Either the propagator $e^{i\mathcal{L}_{fast}\Delta t}$ is applied exactly analytically, or

$e^{i\mathcal{L}_{fast}\Delta t}$ is further factorized with a smaller time step δt that is appropriate for the fast motion.

When the reference system cannot be solved analytically, the RESPA concept can still be applied by introducing a second time step $\delta t = \Delta t/n$ and writing

$$e^{i\mathcal{L}_{fast}\Delta t} == \left\{ \exp\left[\frac{\delta t}{2}\mathbf{F}_{fast}\cdot\frac{\partial}{\partial \mathbf{p}}\right] \exp\left[\delta t\frac{\mathbf{p}}{m}\cdot\frac{\partial}{\partial \mathbf{r}_i}\right] \exp\left[\frac{\delta t}{2}\mathbf{F}_{fast}\cdot\frac{\partial}{\partial \mathbf{p}}\right] \right\}^n \tag{20}$$

Substitution of Eq. 20 into Eq. 19 yields a purely numerical RESPA propagator given by the following set of instructions:

$$\begin{aligned} p &= p + 0.5\Delta t F_{slow} \\ for\ i\ &=\ 1\ to\ n \\ p &= p + 0.5\delta t F_{fast} \\ x &= x + \delta t\frac{p}{m} \\ Recalculate\ &fast\ force \\ p &= p + 0.5\delta t F_{fast} \\ end\ &for \\ Recalculate\ &slow\ force \\ p &= p + 0.5\Delta t F_{slow}. \end{aligned}$$

5 Molecular Dynamics of Hard Spheres (From Allen)

The molecular dynamics of molecules interacting via hard potentials (i.e. discontinuous functions of distance) must be solved in a way which is qualitatively different from the molecular dynamics of soft bodies. Whenever the distance between two particles becomes equal to a point of discontinuity in the potential, then a 'collision' (in a broad sense) occurs: the particle velocities will change suddenly, in a specified manner, depending upon the particular model under study. Thus, the primary aim of a simulation program here is to locate the time, collision partners, and all impact parameters, for every collision occurring in the system, in chronological order. Instead of a regular, step-by-step, approach, as for soft potentials, hard potential programs evolve on a collision-by-collision basis, computing the collision dynamics and then searching for the next collision. The general scheme may be summarized as follows:

- (a) locate next collision;
- (b) move all particles forward until collision occurs;
- (c) implement collision dynamics for the colliding pair;

- (d) calculate any properties of interest, ready for averaging, before returning to (a).

Because of the need to locate accurately future collision times, simulations have been restricted mainly to systems in which force-free motion occurs between collisions, and in which the molecular geometry is spherical. In these simple cases, which include hard spheres [Alder and Wainwright 1959, 1960], rough, and otherwise-modified, hard spheres [O'Dell and Berne 1975; Berne 1977], and square-well molecules [Alder and Wainwright 1959], location of the time of collision between any two particles requires the solution of a quadratic equation. The computational problems become quite daunting when we consider solving the highly non- linear equations that result from models in which the hard cores are supplemented with long-range soft potentials. An example is the primitive model of electrolytes, consisting of hard spheres plus Coulomb interactions. By contrast, such systems may be handled easily using Monte Carlo simulation. Recent developments suggest that it may be possible to treat these 'hybrid' hard + soft systems by returning to an approximate 'step-by- step' approach [Stratt, Holmgren, and Chandler 1981; McNeill and Madden 1982].

A program to solve hard-sphere molecular dynamics has two functions to perform: the calculation of collision times and the implementation of collision dynamics. The collision time calculation is the expensive part of the program, since, in principle, all possible collisions between distinct pairs must be considered. Consider two spheres, i and j, of diameter σ, whose positions at time t are $\mathbf{r}_i$ and $\mathbf{r}_j$, and whose velocities are $\mathbf{v}_i$ and $\mathbf{v}_j$. If these particles are to collide at time $t + \Delta t$ then the following equation will be satisfied:

$$|\mathbf{r}_{ij}(t + \Delta t)| = |\mathbf{r}_{ij}(t) + \Delta t \mathbf{v}_{ij}| = \sigma$$

Defining $b_{ij} = \mathbf{r}_{ij} \cdot \mathbf{v}_{ij}$ the previous equation (second equality) becomes

$$\mathbf{r}_{ij}(t) \cdot \mathbf{r}_{ij}(t) + \Delta t^2 \mathbf{v}_{ij} \cdot \mathbf{v}_{ij} + 2\Delta t b_{ij} - \sigma^2 = 0 \tag{21}$$

This is a quadratic equation in Δt If $b_{ij} > 0$, then the molecules are going away from each other and they will not collide. If $b_{ij} < 0$, it may still be true that the discriminant is negative, in which case eqn 21 has complex roots and again no collision occurs. Otherwise (assuming that the spheres are not already overlapping) two positive roots arise, the smaller of which corresponds to impact.

$$\Delta t = \frac{-b_{ij} - \sqrt{b_{ij}^2 - v_{ij}^2(r_{ij}^2 - \sigma^2)}}{v_{ij}^2} \tag{22}$$

A list of all collision times, for all pairs in the system, allow us to evaluate the shortest of them and propagate the system from t to the shorted found Δt. All molecules are moved forward by the time Δt , the periodic boundary conditions are applied, and the table of future collision times is adjusted accordingly.

Note that in applying the minimum image convention for periodic boundaries one only need to examine the nearest images of any two particles in order to pick out the collision between them. This approximation breaks down a low densities.

Now we are ready to carry through the second part of the calculation, namely the collision dynamics themselves. The changes in velocities of the colliding pair are completely dictated by the requirements that energy and linear momentum be conserved and (for smooth hard spheres) that the impulse acts along the line of centres.

For particles of equal masses, the collision generates only a variation of the component of the velocity along the $\mathbf{r}_{ij}$ direction. Specifically

$$\delta\mathbf{v}_{ij} = -\frac{\mathbf{r}_{ij}\cdot\mathbf{v}_{ij}}{|\mathbf{r}_{ij}|}\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} = -\frac{b_{ij}}{\sigma^2}\mathbf{r}_{ij}$$

so that

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \delta\mathbf{v}_{ij}$$

and

$$\mathbf{v}_j(t+\Delta t) = \mathbf{v}_j(t) - \delta\mathbf{v}_{ij}$$

The event-driven simulations can be implemented also in the case of the square-well potential. Now, for each pair, there are two distances at which 'collisions' occur, so the algorithm for determining collision times is slightly more involved. Collisions at the inner sphere obey normal hard-sphere dynamics; at the outer boundary, the change in momentum is determined by the usual conservation laws. For molecules approaching each other, the potential energy drops on crossing the boundary, and so the kinetic energy shows a corresponding increase. If the molecules are separating, two possibilities arise. If the total kinetic energy is sufficient, the molecules cross the boundary with a loss of kinetic energy to compensate the rise in potential energy. Alternatively, if the kinetic energy is insufficient, reflection at the outer boundary occurs and the particles remain 'bound'. More complicated potentials involving several 'steps' can be treated in the same way; a quite realistic potential can be constructed from a large number of vertical and horizontal segments, but of course the simulation becomes more expensive as more 'collisions' have to be dealt with per unit time [Chapela, Martinez-Casas, and Alejandre 1984].

6 Brownian Simulation - Elaborated from Allen

A straightforward method of conducting 'Brownian dynamics' simulations based on

$$\dot{\mathbf{p}}(t) = -\xi\mathbf{p}(t) + \mathbf{f}(t) + \mathbf{R}(t)$$

(where $\mathbf{R}(t)$ is a random force) has been developed by Ermak [Ermak 1976; Ermak and Buckholtz 1980]. Somewhat different schemes have been employed elsewhere [Morf and

Stoll 1977; Schneider and Stoll 1978; Turq, Lantelme, and Friedman 1977; Adelman 1979]. In Ermak's approach, the equations of motion are integrated over a time interval δt under the assumption that the systematic forces $\mathbf{f}(t)$ remain approximately constant. The result is an algorithm based on stored positions, velocities, and accelerations. For a one-component atomic system, the algorithm may be written

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + c_1 \delta t \mathbf{v}(t) + + c_2 \delta t^2 \mathbf{a}(t) + \delta \mathbf{r}^G$$
$$\mathbf{v}(t+\delta t) = c_0 \mathbf{v}(t) + c_1 \delta t \mathbf{a}(t) + \delta \mathbf{v}^G$$

with

$$c_0 = e^{-\xi \delta t}$$
$$c_1 = \frac{(1-c_0)}{\xi \delta t}$$
$$c_2 = \frac{(1-c_1)}{\xi \delta t}$$

Here

$$\delta \mathbf{r}^G = \int_t^{t+\delta t} \xi^{-1}(1 - e^{-\xi(t+\delta t - t')}) m^{-1} \mathbf{R}(t') dt'$$

and

$$\delta \mathbf{v}^G = \int_t^{t+\delta t} e^{-\xi(t+\delta t - t')} m^{-1} \mathbf{R}(t') dt'$$

Indeed, a formal integration of the velocity equation gives

$$m\mathbf{v}(t) = m\mathbf{v}(0) exp(-\xi t) + \exp(-\xi t) \int_0^t exp(\xi s)(\mathbf{f}(s) + \mathbf{R}(s)) ds \tag{23}$$

and

$$m\mathbf{v}(t+\delta t) = m\mathbf{v}(0) \exp(-\xi(t+\delta t)) + \exp(-\xi(t+\delta t) \int_0^{t+\delta t} \exp(\xi s)(\mathbf{f}(s) + \mathbf{R}(s)) ds \tag{24}$$

such that, substituting $m\mathbf{v}(0) \exp(-\xi t) = m\mathbf{v}(t) - \exp(-\xi t) \int_0^t \exp(\xi s)(\mathbf{f}(s) + \mathbf{R}(s)) ds$

$$m\mathbf{v}(t+\delta t) = m\mathbf{v}(t) exp(-\xi \delta t) + \exp(-\xi(t+\delta t)) \int_t^{t+\delta t} \exp(\xi s)(\mathbf{f}(s) + \mathbf{R}(s)) ds \tag{25}$$

Assuming that the force $\mathbf{f}(s)$ can be approximated as a constant between t and δt, the contribution to the integral becomes

$$\exp(-\xi(t+\delta t))\int_t^{t+\delta t} exp(\xi s)\mathbf{f}(s)ds \approx \exp(-\xi(t+\delta t))\mathbf{f}(t)\int_t^{t+\delta t} exp(\xi s)ds =$$

$$\exp(-\xi(t+\delta t))\mathbf{f}(t)\frac{1}{\xi}[exp(\xi(t+\delta t)) - exp(\xi t) = \frac{\mathbf{f}(t)}{\xi}[1-\exp(-\xi\delta t) = \delta t c_1 \mathbf{f}(t)$$

so that

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t)c_0(\delta t) + \delta t c_1(\delta t)\mathbf{f}(t) + \delta\mathbf{v}^G \tag{26}$$

Indicando δt con s, $\mathbf{v}(t+s) = \mathbf{v}(t)c_0(s) + sc_1(s)\mathbf{f}(t) + \delta\mathbf{v}^G$,

To find the associated evolution of r one integrates the previous solution for v as

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \int_0^{\delta t}\left[(\mathbf{v}(t)c_0(s) + sc_1(s)\mathbf{f}(t))\right]ds + \int_t^{t+\delta t}\delta\mathbf{v}^G ds =$$

$$\mathbf{r}(t) + \int_0^{\delta t}\left[(\mathbf{v}(t)e^{-\xi s} + s\frac{(1-e^{-\xi s})}{\xi s}\mathbf{f}(t))\right]ds + \int_t^{t+\delta t}\delta\mathbf{v}^G ds =$$

$$\mathbf{r}(t) + \mathbf{v}(t)\int_0^{\delta t} e^{-\xi s}ds + \frac{\mathbf{f}(t)}{\xi}\int_0^{\delta t}(1-e^{-\xi s})ds + \int_t^{t+\delta t}\delta\mathbf{v}^G ds =$$

$$\mathbf{r}(t) + \mathbf{v}(t)\frac{(e^{-\xi\delta t}-1)}{\xi} + \frac{\mathbf{f}(t)}{\xi}\int_t^{t+\delta t}(1-e^{-\xi s})ds + \int_t^{t+\delta t}\delta\mathbf{v}^G ds =$$

Now consider (between 0 and δt for simplicity) the last integral

$$\int_0^{\delta t}\delta\mathbf{v}^G(s)ds = \int_0^{\delta t} dse^{-\xi s}\int_0^s e^{\xi s'}m^{-1}\mathbf{R}(s')ds' =$$

by changing the integration limits in the surface integration (so that $s' < s < \delta t$ and $0 < s' < \delta t$)

$$\int_0^{\delta t}\left[\int_{s'}^{\delta t} dse^{-\xi s}\right]e^{\xi s'}m^{-1}\mathbf{R}(s')ds' = \int_0^{\delta t} e^{\xi s'}m^{-1}\mathbf{R}(s')ds'(e^{-\xi\delta t} - e^{-\xi s'})\frac{-1}{\xi} =$$

$$\int_0^{\delta t} m^{-1}\mathbf{R}(s')ds'\frac{1-e^{-\xi(\delta t - s')}}{\xi}$$

so that

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + c_1\delta t\mathbf{v}(t) + +c_2\delta t^2\mathbf{a}(t) + \delta\mathbf{r}^G$$

The two equations (for $\mathbf{r}$ and for $\mathbf{v}$) are stochastic in the sense that they equate the statistical properties of $\delta\mathbf{r}^G$ and $\delta\mathbf{v}^G$ with those of the other terms in the equations. Thus, unlike conventional dynamics, there is no 'unique' trajectory: we can only generate a representative trajectory i.e. produce a realization of the stochastic process. In simulation, each pair of vectorial components of $\delta\mathbf{r}^G$ and $\delta\mathbf{v}^G$ is sampled from a bivariate Gaussian distribution [Chandrasekhar 1943]

$$\rho(\delta r_\alpha^G, \delta v_\alpha^G) = \frac{1}{2\pi\sigma_r\sigma_v(1-c_{rv}^2)^{1/2}} \exp\left\{-\frac{1}{2(1-c_{rv}^2)}\left[\left(\frac{(\delta r_\alpha^G)^2}{\sigma_r^2} + \frac{(\delta v_\alpha^G)^2}{\sigma_v^2} - 2c_{rv}\frac{(\delta r_\alpha^G)(\delta v_\alpha^G)}{\sigma_r\sigma_v}\right)\right]\right\}$$

with zero mean and variances given by

$$\sigma_r^2 = <(\delta r_\alpha^G)^2> = <\int_t^{t+\delta t} \xi^{-1}(1-e^{-\xi(t+\delta t-t')})m^{-1}R_\alpha(t')dt' \int_t^{t+\delta t} \xi^{-1}(1-e^{-\xi(t+\delta t-t")})m^{-1}R_\alpha(t")dt" >$$

$$\frac{2k_BT}{m\xi}\int_t^{t+\delta t}(1-e^{-\xi(t+\delta t-t')})^2dt' = (\delta t)^2\frac{k_BT}{m}(\xi\delta t)^{-1}(2-(\xi\delta t)^{-1}(3-4e^{-\xi\delta t}+e^{-2\xi\delta t}))$$

where we have made use of

$$< R_\alpha(t')R_\alpha(t") >= 2m\xi k_BT\delta(t'-t")$$

and analogously

$$\sigma_v^2 = <(\delta v_\alpha^G)^2> = <\int_t^{t+\delta t} e^{-\xi(t+\delta t-t')}m^{-1}R_\alpha(t')dt' \int_t^{t+\delta t} e^{-\xi(t+\delta t-t")}m^{-1}R_\alpha(t")dt" >$$

$$\frac{2k_BT\xi}{m}\int_t^{t+\delta t}(e^{-\xi(t+\delta t-t')})^2dt' = \frac{k_BT}{m}(1-e^{-2\xi\delta t})$$

and finally

$$c_{rv}\sigma_r\sigma_v = <(\delta r_\alpha^G)(\delta v_\alpha^G)> = <\int_t^{t+\delta t} \xi^{-1}(1-e^{-\xi(t+\delta t-t')})m^{-1}R_\alpha(t')dt' \int_t^{t+\delta t} e^{-\xi(t+\delta t-t")}m^{-1}R_\alpha(t")dt" =$$

$$\frac{2k_BT}{m}\int_t^{t+\delta t}(1-e^{-\xi(t+\delta t-t')})e^{-\xi(t+\delta t-t')}dt' = \delta t\frac{k_BT}{m}(\xi\delta t)^{-1}(1-e^{-\xi\delta t})^2$$

At each stage it is essential that correlated values of $\delta\mathbf{r}^G$ and $\delta\mathbf{v}^G$ are sampled as described above, since they are integrals involving the same random process over the same time interval. Different particles, and different vectorial components, are sampled independently.

Ermak's algorithm is an attempt to treat properly both the systematic dynamic and stochastic elements of the Langevin equation. At low values of the friction coefficient

ξ, the dynamical aspects dominate, and Newtonian mechanics is recovered as $\xi \to 0$. Ermak equations then become a simple Taylor series predictor algorithm. As discussed previously, this is not a particularly accurate method of conducting MD simulations, and the same is true of Brownian dynamics at low friction: what is needed here is a stochastic generalization, with friction, of a predictor-corrector or Verlet-like algorithm. A simple algorithm of this type, which reduces to the velocity Verlet algorithm of Section (3.2.1), is obtained if, on integrating the velocity equation, the systematic force is assumed to vary linearly with time.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + c_1 \delta t \mathbf{v}(t) + c_2 \delta t^2 \mathbf{a}(t) + \delta \mathbf{r}^G$$
$$\mathbf{v}(t+\delta t) = c_0 \mathbf{v}(t) + (c_1 - c_2)\delta t \mathbf{a}(t) + c_2 \delta t \mathbf{a}(t+\delta t) + \delta \mathbf{v}^G$$

After the selection of the random components $\delta \mathbf{r}^G$ and $\delta \mathbf{v}^G$ for a given step, the algorithm is implemented in the usual way. Other Verlet-like algorithms have been proposed [Allen 1980, 1982; van Gunsteren and Berendsen 1982] and, although there is no unique way of generalizing the method, these are all closely reIated to each other and provide a similar measure of improvement over the simple predictor of Ermak, at low friction. At high values of ξ the dynamical aspects become less important, and there is little to choose between the different methods.

If long-time configurational dynamics are of interest, then the momentum variables may be dropped from the equations of motion, in the spirit of time- scale separation implicit in the projection-operator method. The 'position Langevin equation' is a simplified version of equations given by Lax [1966] and by Zwanzig [1969]:

An algorithm for the overdamped case [Ermak and Yeh 1974; Ermak 1975] is

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{D}{k_B T}\mathbf{f}(t)\delta t + \delta \mathbf{r}^G$$

where each component $\delta \mathbf{r}^G$ is chosen independently from a Gaussian distribution with zero mean and variance $< (\delta r_\alpha^G)^2 >= 2D\delta t$. As usual, these equations apply to each component of $\mathbf{r}$. The short-time dynamics generated by these equations are even more unrealistic than those resulting from the Langevin equation. In fact, the method is very much more closely related to the force-bias and smart MC methods than to MD.

6.1 Generate gaussian random variable

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-<x>)^2}{2\sigma^2}}$$

A random number ξ generated from this distribution is related to a number [generated from the normal distribution with zero mean and unit variance by

$$\xi' =< x > +\sigma\xi$$

The problem is reduced to sampling from a normal distribution with zero mean and unit variance. One possible method involves two steps and the generation of two uniform random variates [Box and Muller 1958]:

- (a) generate uniform random variables $\tilde{\xi_1}$ and $\tilde{\xi_2}$ between zero and one.
- (b) calculate $\xi_1 = -\sqrt{-2\ln\tilde{\xi_1}}\cos(2\pi\tilde{\xi_2})$ and $\xi_2 = -\sqrt{-2\ln\tilde{\xi_1}}\sin(2\pi\tilde{\xi_2})$

The numbers ξ_1 and ξ_2 are the desired (independent) normally distributed random numbers.

6.2 Generation of correlated random variable

To generate a brownian motion we need to generate correlated pairs of numbers that are normally distributed. Given two independent normal random a_1 and a_2 with zero means and unit variances, the variables

$$a_1' = \sigma_1 a_1$$
$$a_2' = \sigma_2[c_{12}a_1 + (1 - c_{12}^2)^{1/2}a_2]$$

are sampled from the bivariate Gaussian distribution with zero means, variances σ_1 and σ_2 and correlation coefficient c_{12}.

6.3 General rule

To move from a uniform distribution $p(x)dx = dx$ between zero and one and a desired $p(y)$ one need to find an appropriate transformation $y(x)$ and make use of the conservation of probability

$$|p(y)dy| = |p(x)dx|$$

so that

$$p(y) = p(x)\left|\frac{dx}{dy}\right|$$

For example, if $y = -ln(x)$ then y is distributed between 0 and ∞. In this case $|dy/dx| = |-1/x| = exp(-y)$ so that the random numbers $y = -\ln(x)$ are distributed according to an exponential distribution

$$p(y) = exp(-y)$$

The relation can be extended also to multiple variables and in this case the determinant of the Jacobian of the transformation controls the conservation of probability.

6.4 Check for the gaussian variables

Let us show that the transformation

$$x = -\sqrt{-2\ln\tilde{\xi}_1}\cos(2\pi\tilde{\xi}_2)$$

$$y = -\sqrt{-2\ln\tilde{\xi}_1}\sin(2\pi\tilde{\xi}_2)$$

generates two random variable x and y with gaussian distribution.

First we note that

$$x^2 + y^2 = -2\ln\tilde{\xi}_1$$

such that

$$\tilde{\xi}_1 = e^{-\frac{x^2+y^2}{2}}$$

Conservation of probability states

$$P(\tilde{\xi}_1, \tilde{\xi}_2) = |det\mathbf{J}|P(x, y)$$

where the element of $\mathbf{J}$ are $J_{12} = \frac{\partial x}{\partial\tilde{\xi}_1}$ and so on such that

$$det\mathbf{J} = \frac{\partial x}{\partial\tilde{\xi}_1}\frac{\partial y}{\partial\tilde{\xi}_2} - \frac{\partial x}{\partial\tilde{\xi}_2}\frac{\partial y}{\partial\tilde{\xi}_1}$$

We find with some algebra

$$\frac{\partial x}{\partial\tilde{\xi}_1} = -\frac{1}{2}(-2\ln\tilde{\xi}_1)^{-1/2}\frac{(-2)}{\tilde{\xi}_1}\cos(2\pi\tilde{\xi}_2)$$

and

$$\frac{\partial x}{\partial\tilde{\xi}_2} = 2\pi\sqrt{-2\ln\tilde{\xi}_1}\sin(2\pi\tilde{\xi}_2)$$

and analogous relations for y so that

$$det\mathbf{J} = -\frac{1}{2}(-2\ln\tilde{\xi}_1)^{-1/2}\frac{(-2)}{\tilde{\xi}_1}\cos(2\pi\tilde{\xi}_2)2\pi\sqrt{-2\ln\tilde{\xi}_1}\cos(2\pi\tilde{\xi}_2) +$$

$$-\frac{1}{2}(-2\ln\tilde{\xi}_1)^{-1/2}\frac{(-2)}{\tilde{\xi}_1}\sin(2\pi\tilde{\xi}_2)2\pi\sqrt{-2\ln\tilde{\xi}_1}\sin(2\pi\tilde{\xi}_2) = -\frac{2\pi}{\tilde{\xi}_1}$$

Thus $|det\mathbf{J}| = \frac{2\pi}{e^{-\frac{x^2+y^2}{2}}}$. Since $P(\tilde{\xi}_1, \tilde{\xi}_2) = 1$, being $\tilde{\xi}_1$ and $\tilde{\xi}_2$ uniformly distributed between 0 and 1,

$$P(x, y) = \frac{1}{|det\mathbf{J}|} = \frac{e^{-\frac{x^2+y^2}{2}}}{2\pi} = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}\frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}$$

7 Constraints - Tuckerman

In mechanics, it is often necessary to treat a system that is subject to a set of externally imposed constraints. These constraints can be imposed as a matter of convenience, e.g. constraining high-frequency chemical bonds in a molecule at fixed bond lengths, or as true constraints that might be due, for example, to the physical boundaries of a system or the presence of thermal or barostatic control mechanisms. Constraints are expressible as mathematical relations among the phase space variables. Thus, a system with N_c constraints will have $3N-N_c$ degrees of freedom and a set of N_c functions of the coordinates and velocities that must be satisfied by the motion of the system. Constraints are divided into two types. If the relationships that must be satisfied along a trajectory are functions of only the particle positions $\mathbf{r_N}$ and possibly time, then the constraints are called holonomic and can be expressed as N_c conditions of the form

$$\sigma_k(q_1....q_N,t)=0 \quad k=1,....N_c$$

In general, it would seem that the imposition of constraints no longer allows the equations of motion to be obtained from the stationarity of the action, since the coordinates (and/or velocities) are no longer independent. More specifically, the path displacements δq are no longer independent. In fact, the constraints can be built into the action formalism using the method of Lagrange undetermined multipliers. However, in order to apply this method, the constraint conditions must be expressible in a differential form as:

$$\sum_{\alpha=1}^{N} a_{k,\alpha}dq_\alpha + a_{k,t}dt = 0 \quad k=1,....N_c$$

where $a_{k,\alpha}$ is a set of coefficients for the displacements δq_α. For a holonomic constraint it is clear that the coefficients can be obtained by differentiating the constraint condition

$$\frac{\partial\sigma_k(q_1....q_N,t)}{\partial q_\alpha}dq_\alpha + \frac{\partial\sigma_k(q_1....q_N,t)}{\partial t}dt = 0 \quad k=1,....N_c$$

Assuming that the constraints can be expressed in the differential form we must also be able to express them in terms of path displacement δq_α in order to incorporate them into the action principle. Unfortunately, doing so requires a further restriction, since it is not possible to guarantee that a perturbed path $Q(t)+\delta Q(t)$ satisfies the constraints. The latter will hold if the constraints are integrable, in which case they are expressible in terms of path displacements as (e.g. if the constraint does not explicitly changes with time)

$$\sum_{\alpha=1}^{N} a_{k,\alpha}dq_\alpha = 0 \quad k=1,....N_c$$

The equations of motion can then be obtained by adding a set of Lagrange undetermined multipliers, λ_k, where there is one multiplier for each constraint, in the action integral:

$$\delta A = \int_{t_i}^{t_2} \sum_{\alpha=1}^{N} \left[-\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_\alpha}\right) + \frac{\partial \mathcal{L}}{\partial q_\alpha} + \sum_{k=1}^{N_c} \lambda_k a_{k\alpha} \right] \delta q_\alpha(t) dt$$

The equations of motion obtained by requiring that $\delta A = 0$ are then

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_\alpha}\right) - \frac{\partial \mathcal{L}}{\partial q_\alpha} = \sum_{k=1}^{N_c} \lambda_k a_{k\alpha} \tag{27}$$

It may seem that we are still relying on the independence of the displacements δq_α, but this is actually not the case. Suppose we choose the first $3N - N_c$ coordinates to be independent. Then, these coordinates can be evolved using eqns. 27. However, we can choose λ_k such that eqns. 27 apply to the remaining N_c coordinates as well. In this case, eqns. 27 hold for all $3N$ coordinates provided they are solved subject to the constraint conditions. The latter can be expressed as a set of N_c differential equations of the form

$$\sum_{\alpha=1}^{3N} a_{k\alpha} \dot{q}_\alpha = 0 \tag{28}$$

Eqns. 27 together with eqns. 28 constitute a set of $3N - N_c$ equations for the $3N + N_c$ unknowns, $q_1, ..., q_{3N}, \lambda_1, ..., \lambda_{N_c}$. This is the most common approach used in numerical solutions of classical-mechanical problems. Note that, even if a system is subject to a set of time-independent holonomic constraints, the Hamiltonian is still conserved. In order to see this, note that eqns. 27 and 28 can be cast in Hamiltonian form as

$$\dot{q}_\alpha = \frac{\partial \mathcal{H}}{\partial p_\alpha}$$

(remember $H(p,q) = p\dot{q} - L(q,\dot{q})$ so that $\partial H/\partial p = \dot{q}$)

$$\frac{dp_\alpha}{dt} = -\frac{\partial \mathcal{H}}{\partial q_\alpha} + \sum_{k=1}^{N_c} \lambda_k a_{k\alpha}$$

(remember $L = p\dot{q} + H(p,q)$ and $\frac{\partial \mathcal{L}}{\partial \dot{q}} = p$ and $\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial \mathcal{H}}{\partial q}$, so that Eq. 27 transform in the previous equation)

These two relations are associated with the constraints (substituting $\dot{q}$)

$$\sum_{\alpha=1}^{3N} a_{k\alpha} \frac{\partial \mathcal{H}}{\partial p_\alpha} = 0$$

Computing the time-derivative of the Hamiltonian, we obtain

$$\frac{d\mathcal{H}}{dt} = \sum_\alpha \left[\frac{\partial \mathcal{H}}{\partial q_\alpha}\dot{q}_\alpha + \frac{\partial \mathcal{H}}{\partial p_\alpha}\dot{p}_\alpha \right] = \sum_\alpha \left[\frac{\partial \mathcal{H}}{\partial q_\alpha}\frac{\partial \mathcal{H}}{\partial p_\alpha} + \frac{\partial \mathcal{H}}{\partial p_\alpha}\left(-\frac{\partial \mathcal{H}}{\partial q_\alpha} + \sum_{k=1}^{N_c} \lambda_k a_{k\alpha} \right) \right] =$$

$$\sum_\alpha \frac{\partial \mathcal{H}}{\partial p_\alpha} \sum_{k=1}^{N_c} \lambda_k a_{k\alpha} = \sum_{k=1}^{N_c} \lambda_k \sum_\alpha \frac{\partial \mathcal{H}}{\partial p_\alpha} a_{k\alpha} = 0$$

From this, it is clear that no work is done on a system by the imposition of holonomic constraints.

7.1 one particle case

The constrained equations of motion Eqns. 27 together with eqns. 28 constitute a complete set of equations for the motion subject to the N_c constraint conditions. Let us study these equations in more detail. For the purposes of this discussion, consider a single particle in three dimensions described by a single Cartesian position vector $\mathbf{r}$(t) subject to a single constraint $\sigma(\mathbf{r}) = 0$. According to eqns. 27 and eqns. 28, the constrained equations of motion take the form

$$\begin{aligned} m\ddot{\mathbf{r}} &= \mathbf{F}(\mathbf{r}) + \lambda \nabla\sigma \\ \nabla\sigma \cdot \dot{\mathbf{r}} &= 0 \end{aligned} \tag{29}$$

These equations will generate classical trajectories of the system for different initial conditions $\mathbf{r}(\mathbf{0}), \dot{\mathbf{r}}(\mathbf{0})$ provided the condition $\sigma(\mathbf{r}(\mathbf{0})) = 0$ is satisfied. If this condition is true, then the trajectory will obey $\sigma(\mathbf{r}(t)) = 0$. Conversely, for each $\mathbf{r}$ visited along the trajectory, the condition $\sigma(\mathbf{r}) = 0$ will be satisfied. The latter condition defines a surface on which the motion described by eqns. (29 must remain. This surface is called the surface of constraint. The quantity $\nabla\sigma(\mathbf{r})$ is a vector that is orthogonal to the surface at each point $\mathbf{r}$. Thus, the second equation 29 expresses the fact that the velocity must also lie in the surface of constraint, hence it must be perpendicular to $\nabla\sigma(\mathbf{r})$. Of the two force terms appearing in eqns. 29, the first is an "unconstrained" force which, alone, would allow the particle to drift off of the surface of constraint. The second term must, then, correct for this tendency. If the particle starts from rest, this second term exactly removes the component of the force perpendicular to the surface of constraint. This minimal projection of the force, first conceived by Karl Friedrich Gauss (1777-1855), is known as Gauss's principle of least constraint (Gauss, 1829). The component of the force perpendicular to the surface is

$$\mathbf{F}_\perp = [\mathbf{n}(\mathbf{r}) \cdot \mathbf{F}(\mathbf{r})]\mathbf{n}(\mathbf{r})$$

where $\mathbf{n}(\mathbf{r})$ is a unit vector perpendicular to the surface at $\mathbf{r}$; $\mathbf{n}(\mathbf{r})$ is given by

$$\mathbf{n}(\mathbf{r}) = \frac{\nabla\sigma(\mathbf{r})}{|\nabla\sigma(\mathbf{r})|}$$

Thus, the component of the force parallel to the surface is

$$\mathbf{F}_{||} = \mathbf{F}(\mathbf{r}) - \mathbf{F}_\perp = \mathbf{F}(\mathbf{r}) - [\mathbf{n}(\mathbf{r}) \cdot \mathbf{F}(\mathbf{r})]\mathbf{n}(\mathbf{r})$$

If the particle is not at rest, the projection of the force cannot lie entirely in the surface of constraint. Rather, there must be an additional component of the projection which can project any free motion of the particle directed off the surface of constraint. This additional term must sense the curvature of the surface in order to affect the required projection; it must also be a minimal projection perpendicular to the surface.

8 Shake and Rattle in theory (Tuckermann)

For time-independent holonomic constraints σ_k, the Lagrangian formulation of the equations of motion in Cartesian coordinates are

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}_i}\right) - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{r}_i}\right) = \sum_1^{N_c} \lambda_k \mathbf{a}_{ki} \tag{30}$$

and (the derivative of the constraint equal to zero, so that the constraint if true at time 0 is always true)

$$\sum_{i=1}^{N} \mathbf{a}_{ki} \cdot \dot{\mathbf{r}}_i = 0$$

where

$$\mathbf{a}_{ki} = \nabla_i \sigma_k(\mathbf{r}_1 .... \mathbf{r}_N)$$

Note that these equations are equivalent to (remember $\mathcal{L}(\mathbf{r}, \dot{\mathbf{r}}) = T - V$, $\partial \mathcal{L}/\partial v_{i,\alpha} = m v_{i,\alpha}$ and $\partial \mathcal{L}/\partial r_{i,\alpha} = -\partial V/\partial r_{i,\alpha} = F_{i,\alpha}$)

$$m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i + \sum_{k=1}^{N_c} \lambda_k \nabla_i \sigma_k$$

and

$$\frac{d}{dt}\sigma_k(\mathbf{r}_1 .... \mathbf{r}_N) = 0$$

The constraint problem amounts to integrating eqn. 30 subject to the conditions $\sigma_k(\mathbf{r}_1 .... \mathbf{r}_N) = 0$ and $\dot{\sigma}_k(\mathbf{r}_1 .... \mathbf{r}_N) = \sum_{i=1}^{N} \nabla_i \sigma_k(\mathbf{r}_1 .... \mathbf{r}_N) \cdot \dot{\mathbf{r}}_i = 0$.

We wish to develop a numerical scheme in which the constraint conditions are satisfied exactly as part of the integration algorithm. Starting from the velocity Verlet approach, for example, we begin with the position update, which, when holonomic constraints are imposed, reads

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0) + \Delta t \mathbf{v}_i(0) + \frac{\Delta t^2}{2m}\mathbf{F}_i(0) + \frac{\Delta t^2}{2m}\sum_k \lambda_k \nabla_i \sigma_k(0) \tag{31}$$

where $\sigma_k(0) = \sigma_k(\mathbf{r}_1(0) .... \mathbf{r}_N(0))$. In order to ensure that the constraint is satisfied exactly at time Δt , we impose the constraint condition directly on the numerically obtained

positions $\mathbf{r}_i(\Delta t)$ and determine, on the fly, the multipliers needed to enforce the constraint. Let us define

$$\mathbf{r}_i^{'} = \mathbf{r}_i(0) + \Delta t \mathbf{v}_i(0) + \frac{\Delta t^2}{2m}\mathbf{F}_i(0) \tag{32}$$

so that

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i^{'} + \sum_k \tilde{\lambda}_k \nabla_i \sigma_k(0) \tag{33}$$

where $\tilde{\lambda}_k = \frac{\Delta t^2}{2m}\lambda_k$. Then, for each constraint condition we impose

$$\sigma_l \left(\mathbf{r}_1^{'} + \sum_k \tilde{\lambda}_k \nabla_1 \sigma_k(0), \quad ....., \quad \mathbf{r}_N^{'} + \sum_k \tilde{\lambda}_k \nabla_N \sigma_k(0) \right) = 0 \tag{34}$$

Unless the constraints are of a particularly simple form will need to be solved iteratively. A simple procedure for doing this is, known as the SHAKE algorithm (Ryckaert et al., 1977), proceeds as follows. First, if a good initial guess of the solution, $\lambda_k^{(1)}$, is available (for example, the multipliers from the previous molecular dynamics time step), then the coordinates can be updated according to

$$\mathbf{r}_i^{(1)} = \mathbf{r}_i^{'} + \sum_k \tilde{\lambda}_k^{(1)} \nabla_i \sigma_k(0) \tag{35}$$

The exact solution for the multipliers is now written as $\tilde{\lambda}_k^{(1)} + \delta\tilde{\lambda}_k^{(1)}$ and

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i^{(1)} + \sum_k \delta\tilde{\lambda}_k^{(1)} \nabla_i \sigma_k(0) \tag{36}$$

so that eqn. 34 becomes

$$\sigma_l \left(\mathbf{r}_1^{(1)} + \sum_k \delta\tilde{\lambda}_k^{(1)} \nabla_1 \sigma_k(0), \quad ....., \quad \mathbf{r}_N^{(1)} + \sum_k \delta\tilde{\lambda}_k \nabla_N \sigma_k(0) \right) = 0 \tag{37}$$

Next, eqn. 37 is expanded to first order in a Taylor series about $\tilde{\lambda}_k^{(1)} = 0$.

$$\sigma_l \left(\mathbf{r}_i^{(1)}, \quad ....., \quad \mathbf{r}_N^{(1)} \right) + \sum_{i=1}^{N} \sum_{k=1}^{N_c} \nabla_i \sigma_l(\mathbf{r}_1^{(1)}, \quad ....., \quad \mathbf{r}_N^{(1)}) \nabla_i \sigma_k(\mathbf{r}_1(0), ...\mathbf{r}_N(0)) \delta\tilde{\lambda}_k^{(1)} \approx 0 \tag{38}$$

Eqn. 38 is a matrix equation for the changes $\delta\tilde{\lambda}_k^{(1)}$ If the dimensionality of this equation is not too large, then it can be inverted directly to yield the full set of $\delta\tilde{\lambda}_k^{(1)}$ simultaneously. This procedure is known as matrix-SHAKE or M-SHAKE (Kraeutler et al., 2001).

Because eqn. Eqn. 38 was approximated by linearization, however, adding the correction $\sum_k \delta\tilde{\lambda}_k^{(1)} \nabla_i \sigma_k(0)$ to $\mathbf{r}_i^{(1)}$ does not yield a fully converged $\mathbf{r}_i(\Delta t)$.

We, therefore, define

$$\mathbf{r}_i^{(2)} = \mathbf{r}_i^{(1)} + \sum_k \tilde{\delta}\lambda_k^{(1)} \nabla_i \sigma_k(0) \tag{39}$$

and

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i^{(2)} + \sum_k \delta\tilde{\lambda}_k^{(2)} \nabla_i \sigma_k(0) \tag{40}$$

The procedure is repeated until the constraint conditions are satisfied to a given small tolerance.

Once the multipliers are obtained, and the coordinates fully updated, the velocities must be updated as well to make sure that the constraints $\dot{\sigma}_k(\mathbf{r}_1, ...\mathbf{r}_N) = 0$ (for all k) are satisfied. The velocities can be evaluated in two steps (an equivalent formulation of the velocity Verlet algorithm)

First one calculate the terms depending on the positions at time 0 (for which the constraints have been already calculated)

$$\mathbf{v}_i(\Delta t/2) = \mathbf{v}_i(0) + \frac{\Delta t/2}{m}\mathbf{F}_i(0) + \frac{1}{\Delta t}\sum_k \tilde{\lambda}_k \nabla_i \sigma_k(0)$$

and once the new forces are calculated from the updated positions then one evaluates the constraints at time Δt (note indeed that the ∇ of the constraints is now evaluated at Δt) this time requiring

$$\mathbf{v}_i(\Delta t) = \mathbf{v}_i(\Delta t/2) + \frac{\Delta t/2}{m}\mathbf{F}_i(\Delta t) + \frac{\Delta t}{2m}\sum_k \mu_k \nabla_i \sigma_k(\Delta t) = \mathbf{v'}_i + \frac{1}{m}\sum_k \tilde{\mu}_k \nabla_i \sigma_k(\Delta t)$$

where μ_k has been used to denote the multiplier for the velocity step to indicate that they are different from those used for the position step and $\tilde{\mu}_k = (\Delta t/2)\mu_k$. The multiplier μ_k are now obtained enforcing the condition

$$\sum_i \nabla_i \sigma(\Delta t) \cdot \mathbf{v}_i(\Delta t) = 0$$

on the velocities. Substituting in these equationd $\mathbf{v}_i(\Delta t)$ we obtain a set of N_c linear equations for the multiplier $\tilde{\mu}_k$

$$\sum_i \nabla_i \sigma(\Delta t) \cdot \left(\mathbf{v}'_i + \frac{\Delta t}{2m}\sum_k \tilde{\mu}_k \nabla_i \sigma_k(\Delta t) \right) = 0$$

If these equations are solved iteratively, similarly to what done for the spatial constraints, the dynamic of a system with holonomic constraints can be generated in the velocity Verlet scheme. This algorithm is called RATTLE.

8.1 Evolution of a dimer in the absence of an external force - example

Let us assume we have a dimer defined by coordinated $\mathbf{r_1}$ and $\mathbf{r_2}$ and that the relative distance between the two particles is d, e.g.

$$\sigma \equiv (\mathbf{r_1} - \mathbf{r_2})^2 - d^2 = 0$$

We have thus a system with two particles and one constraint only. In the Lagrangian formalism, we have to solve

$$m\ddot{\mathbf{r}}_i = \mathbf{F}_i + \lambda \nabla_i \left[(\mathbf{r_1} - \mathbf{r_2})^2 - d^2 \right]$$

and

$$\frac{d}{dt}\sigma = \frac{d}{dt}\left[(\mathbf{r_1} - \mathbf{r_2})^2 - d^2 \right] = \sum_{i=1}^{2} \nabla_i \left[(\mathbf{r_1} - \mathbf{r_2})^2 - d^2 \right] \cdot \dot{\mathbf{r}}_i = 0$$

The velocity Verlet, for $\mathbf{F} = 0$ gives

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0) + \Delta t \mathbf{v}_i(0) + \frac{\Delta t^2}{2m}\lambda \nabla_i \left[(\mathbf{r_1} - \mathbf{r_2})^2 - d^2 \right] = \mathbf{r}_i^{'} - \frac{\Delta t^2}{2m}\lambda 2(-1)^i(\mathbf{r_1} - \mathbf{r_2}) \quad (41)$$

Substituting in the constraint equation one obtains (defining $\mathbf{r}' \equiv \mathbf{r}_i(0) + \Delta t \mathbf{v}_i(0)$)

$$\left[\mathbf{r}_1^{'} + 2\frac{\Delta t^2}{2m}\lambda(\mathbf{r_1} - \mathbf{r_2}) - \mathbf{r}_2^{'} + 2\frac{\Delta t^2}{2m}\lambda(\mathbf{r_1} - \mathbf{r_2}) \right]^2 = d^2$$

that provides a quadratic expression for λ that can be solved and the position step of velocity Verlet (Eq. 41) can be properly calculated.

The velocities need to be evaluated in two steps

First,

$$\begin{aligned} \mathbf{v}_i(\Delta t/2) &= \mathbf{v}_i(0) + \frac{\Delta t/2}{m}\mathbf{F}_i(0) + \frac{\Delta t/2}{m}\lambda \nabla_i \left[(\mathbf{r_1}(0) - \mathbf{r_2}(0))^2 - d^2 \right] \\ &= \mathbf{v}_i^{'} - \frac{\Delta t/2}{m}\lambda 2(-1)^i(\mathbf{r_1}(0) - \mathbf{r_2}(0)) \end{aligned}$$

and once the new forces are calculated from the updated positions (note that the ∇ of the constraints are now evaluated at Δt)

$$\begin{aligned} \mathbf{v}_i(\Delta t) &= \mathbf{v}_i(\Delta t/2) + \frac{\Delta t/2}{m}\mathbf{F}_i(\Delta t) + \frac{\Delta t/2}{m}\mu \nabla_i \left[(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t))^2 - d^2 \right] \\ &= \mathbf{v}_i^{'} - \frac{\Delta t/2}{m}\mu 2(-1)^i(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t)) \end{aligned}$$

and this time the multiplier μ are calculated from

$$\sum_i \nabla_i \sigma(\Delta t) \cdot \mathbf{v}_i(\Delta t) = 0$$

which in the present case gives

$$2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t)) \cdot [\mathbf{v}_1^{'}(\Delta t) + \frac{\Delta t/2}{m} \mu 2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t))] +$$

$$-2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t)) \cdot [\mathbf{v}_2^{'}(\Delta t) - \frac{\Delta t/2}{m} \mu 2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t))] = 0$$

or

$$\mu = \frac{2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t)) \cdot [\mathbf{v}_1^{'}(\Delta t) - \mathbf{v}_2^{'}(\Delta t)]}{\frac{\Delta t/2}{m} 4(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t)) \cdot [2(\mathbf{r_1}(\Delta t) - \mathbf{r_2}(\Delta t))]}$$

This value of μ can be used to update the velocity according to the velocity Verlet.