Supplementary Information

A GPU-accelerated immersive audio-visual framework for Interaction with molecular dynamics using consumer depth sensors

David R. Glowacki[1,2,3,4], Michael O'Connor,[3] Gaetano Calabró,[5] James Price,[3] Philip Tew,[2] Thomas Mitchell,[2,6] Joseph Hyde,[7] David P. Tew,[1] David J. Coughtrie,[1] and Simon McIntosh-Smith[3]

[1]School of Chemistry, University of Bristol, Bristol, BS8 1TS, UK
[2]Pervasive Media Studio, Watershed, 1 Canons Rd, Bristol BS1 5TX
[3]Department of Computer Science, University of Bristol, BS2 9LG, UK
[4]Department of Chemistry, Stanford University, Stanford, CA, 94305, USA
[5]EaStCHEM School of Chemistry, University of Edinburgh, Edinburgh EH9 3JJ
[6]Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY
[7]School of Music and Performing Arts, Bath Spa University, Bath BA2 9BN

Modified Berendsen Thermostat

Standard dynamics propagation schemes (utilizing reasonably sized timesteps), which usually rely on being able to express the force as a low-order Taylor series expansion, are not always well-equipped to deal with the ill-behaved forces that arise from an interactive simulation. This can lead to explosions in the dynamical propagation as a consequence of rapid numerical error accumulation. To address this, and avoid the numerical explosions associated with such instabilities, we have implemented a modified velocity rescaling Berendsen thermostat, in which the instantaneous system temperature T_t approaches some desired temperature T_0 with a first order rate

$$\frac{dT_t}{dt} = \frac{1}{\tau} \cdot (T_0 - T_t) \tag{S1}$$

that depends on a user-specified rate coefficient ($1/\tau$) and how far the system is from T_0. Rearranging (S1) gives an expression for the temperature change dT_t over some time step dt:

$$dT_t = \frac{dt}{\tau} \cdot (T_0 - T_t) \tag{S2}$$

where τ is a first order time constant, and

$$T_t = \frac{1}{d \cdot N \cdot k_B} \sum_{i=1}^{N} m_i v_i^2 \tag{S3}$$

with d the number of dimensions in which each atom can move (three), N the number of atoms in the simulation, and k_B the Boltzmann constant. The velocity rescaling constant λ is determined via definition of $T(\lambda)$, which is the temperature that results when all the atomic velocities are scaled by λ, i.e.:

$$T(\lambda) = \frac{1}{df \cdot N \cdot k_B} \left[\sum_{i=1}^{N} m_i (\lambda v_i)^2 \right] = \lambda^2 T_t \tag{S4}$$

λ is evaluated by specifying that $dT_t = T(\lambda) - T_t$, and substituting Eq (S2) and (S4) to give:

$$\frac{dt}{\tau} \cdot (T_0 - T_t) = \lambda^2 T_t - T_t \qquad \text{(S5)}$$

which may be solved to yield

$$\lambda = \sqrt{1 + \frac{dt}{\tau}\left[\frac{T_0}{T_t} - 1\right]} \qquad \text{(S6)}$$

Eq (S1) – (S6) are the standard Berendsen velocity rescaling scheme; but we found them to be unreliable for ensuring the stability of dS when exposed to users. Stability was considerably improved by looping over the atomic velocities to ensure that none of the atoms within the simulation have a velocity more than two standard deviations larger than the average atomic velocity (prior to determining the value of T_t required for calculating the atomic velocity scale factor λ). This procedure then gives a good compromise between computational efficiency, interactive fluidity, and system stability. It eliminates numerical instabilities that can arise when user motion suddenly 'injects' energy into the system Hamiltonian.