new configuration at $t+h$ satisfies the constraints. Thus the SHAKE algorithm can be written as

$$r(t+h) = 2r(t) - r(t-h) + h^2[f[r(t)] + g_S(t)],$$

where g_S is the SHAKE approximation for g.

SHAKE has the same disadvantages as those associated with the Verlet algorithm. It might be thought that a "velocity version" of SHAKE could eliminate these disadvantages. However, such a formulation of the equations of motion for constrained dynamics would give

$$r(t+h) = r(t) + h\dot{r}(t) + h^2[f[r(t)] + g[r(t), \dot{r}(t)]]/2, \tag{2.5}$$

$$\dot{r}(t+h) = \dot{r}(t) + h[f[r(t)] + g[r(t), \dot{r}(t)] + f[r(t+h)] + g[r(t+h), \dot{r}(t+h)]]/2. \tag{2.6}$$

Suppose we know $r(t)$ and $\dot{r}(t)$ and we want to solve these equations by iteration using the ideas of SHAKE to evaluate the unknown forces g. Then using (2.5) we could calculate $r(t+h)$ by replacing $g[r(t), \dot{r}(t)]$ by an approximation that made $r(t+h)$ satisfy the constraints. But then we cannot use (2.6) to get $\dot{r}(t+h)$ because we have no way of evaluating the second g that appears in that equation. In other words, by (2.5), we need to know $\dot{r}(t)$ before beginning the calculation of $g(t)$, but by (2.6) we need to know $g(t+h)$ before calculating $\dot{r}(t+h)$. This is inconsistent with a simple iterative scheme.

There is a straightforward way of eliminating this difficulty. There is no need that the same approximation for g be used in the position equation as in the velocity equation. This suggests the following procedure. Suppose we know the positions, forces, and velocities at time t. Using (2.5) we calculate $r(t+h)$ by choosing the g that appears in this equation so that $r(t+h)$ satisfies the constraints exactly (or to within a desired precision). Thus we replace (2.5) by

$$r(t+h) = r(t) + h\dot{r}(t) + h^2[f[r(t)] + g_{RR}(t)]/2. \tag{2.7}$$

Knowing $r(t+h)$, we can calculate $f[r(t+h)]$. Then using (2.6) we calculate $\dot{r}(t+h)$ by choosing the second g that appears in this equation so that the resulting $\dot{r}(t+h)$ satisfies the time dericatives of the constraints exactly (or within a desired tolerance). Thus we replace (2.6) by

$$\dot{r}(t+h) = \dot{r}(t) + h[f[r(t)] + g_{RR}(t) + f[r(t+h)] + g_{RV}(t)]/2. \tag{2.8}$$

This algorithm, which will be called RATTLE, makes two separate approximations, g_{RR} and g_{RV}, for the forces associated with the constraints. As a result, it is possible to require that both the positions and the velocities satisfy the constraints. The detailed equations for RATTLE are given in Appendix A.