1 Leap-frog algorithm

The Verlet approximation is

\[ e^{iL \Delta t} \approx e^{iL_p \Delta t/2} e^{iL_q \Delta t} e^{iL_p \Delta t/2}. \]

The leapfrog factorization is obtained by performing a similar approximation

\[ e^{iL \Delta t} \approx e^{iL_q \Delta t/2} e^{iL_p \Delta t} e^{iL_q \Delta t/2}. \]

Let us compute the different transformations:

a) Applying \( e^{iL_q \Delta t/2} \):

\[ e^{iL_q \Delta t/2} \left( \begin{array}{c} p \\ q \end{array} \right) = \left( \begin{array}{c} p \\ q + \frac{\Delta t}{2m} p \end{array} \right) \]

b) Applying \( e^{iL_p \Delta t} \):

\[ e^{iL_p \Delta t} \left( \begin{array}{c} p \\ q + \frac{\Delta t}{2m} p \end{array} \right) = \left( \begin{array}{c} p + F(q) \Delta t \\ q + \frac{\Delta t}{2m} [p + F(q) \Delta t] \end{array} \right) \]

c) Finally, we apply \( e^{iL_q \Delta t/2} \) again

\[ e^{iL_q \Delta t/2} \left( \begin{array}{c} p \cdot F(q) \Delta t \\ q + \frac{\Delta t}{2m} [p + F(q) \Delta t] \end{array} \right) = \left( \begin{array}{c} p + F \left( q + \frac{\Delta t}{2m} p \right) \Delta t \\ q + \frac{\Delta t}{2m} [p + F \left( q + \frac{\Delta t}{2m} p \right) \Delta t] \end{array} \right) \]

Let us now apply this transformation in practice. Suppose we know \( q(t) \) and \( p(t) \). In the leapfrog scheme, we define a value of \( q \) that, conventionally, we associate to time \( t + \Delta t/2 \):

\[ q(t + \Delta t/2) = q(t) + \frac{\Delta t}{2m} p(t) \]

Then, we have

\[
\begin{align*}
p(t + \Delta t) &= p(t) + F \left( q(t) + \frac{\Delta t}{2m} p(t) \right) \Delta t = p(t) + \Delta t F[q(t + \Delta t/2)] \\
q(t + \Delta t) &= q(t) + \frac{\Delta t}{m} p(t) + \frac{\Delta t}{2m} F \left( q(t) + \frac{\Delta t}{2m} p(t) \right) \Delta t = q(t + \Delta t/2) + \frac{\Delta t}{2m} p(t + \Delta t) \end{align*}
\]

In the following step, we start again by computing

\[ q(t + \frac{3}{2} \Delta t) = q(t + \Delta t) + \frac{\Delta t}{2m} p(t + \Delta t) = q(t + \Delta t/2) + \frac{\Delta t}{m} p(t + \Delta t). \]

Note that we do not need \( q(t + \Delta t) \) to compute \( q(t + \frac{3}{2} \Delta t) \); only \( q(t + \Delta t/2) \) is needed. We can thus use an algorithm that does not compute \( q(t + \Delta t) \). Thus the algorithm computes:

\[ q(t + \Delta t/2) \to p(t + \Delta t) \to q(t + 3\Delta t/2) \to p(t + 2\Delta t) \to q(t + 5\Delta t/2) \to \ldots \]

Note that the leapfrog relations can be rewritten as

\[
\begin{align*}
p(t) &= m \underbrace{\left[ q(t + \Delta t/2) - q(t - \Delta t/2) \right]}_{\Delta t} \\
F[q(t + \Delta t/2)] &= \underbrace{\left[ p(t + \Delta t) - p(t) \right]}_{\Delta t}
\end{align*}
\]
where one immediately recognizes a simple discrete version of the derivatives of $q$ and $p$ with respect to time. In this naive derivation of the leapfrog recursions, it is quite natural to associate the values of $q$ to the midpoints $n\Delta t + \Delta t/2$.

**Practical implementation with starting values $q_0$ and $p_0$:**

\begin{verbatim}
q[0] = q0; p[0] = p0; q[1] = q0 + Deltat*p0/(2*m);
for i = 1,....., Number_of_iterations
    F = F(q[i]);
    p[i] = p[i-1] + Deltat*F;
    q[i+1] = q[i] + Deltat*p[i]/m;
endfor
\end{verbatim}

Here $q[n]$ is the value of $q$ at time $(n - 1/2)\Delta t$, and $p[n]$ is the value of $p$ at time $n\Delta t$. If needed, we can compute $q(n\Delta t)$, using Eq. (1).