Accuracy of the Simple Integration Scheme

A numerical integration scheme such as the one just described solves an approximation to the true differential equation. Thus, it necessarily make *errors*—deviations between the numerical and the true solution that have nothing to do with mistakes in the program, nor misconceptions on the part of the programmer. Errors are part and parcel of the approximations underlying the scheme.

A numerical integration scheme is equivalent to writing down as many terms as possible of the *Taylor series expansion* of the solution $x(t)$ about the point t_i

$$\begin{aligned} x(t_i + \delta t) &= x(t_i) + x'(t_i)\,\delta t + \tfrac{1}{2}\,x''(t_i)\,\delta t^2 + \tfrac{1}{6}\,x'''(t_i)\,\delta t^3 + \ldots + \tfrac{1}{n!}\,x^{(n)}(t_i)\,\delta t^n + \ldots \\ &= x_i + v_i\,\delta t + \tfrac{1}{2}\,a_i\,\delta t^2 + \tfrac{1}{6}\,a_i'\,\delta t^3 + \ldots \end{aligned}$$

In our case, since we approximate $\mathbf{a}$ by a constant for the duration of each step, if $\mathbf{a}$ is differentiable is is clear that the error we incur in $\mathbf{a}$ is $O(\delta t)$—actually $\mathbf{a}'\delta t$, the next term in the Taylor series expansion of $\mathbf{a}$, which we explicitly throw away. The corresponding errors in $\mathbf{x}$ and $\mathbf{v}$ at the end of the step are evidently $O(\delta t^3)$ and $O(\delta t^2)$, respectively.

The lowest power of δt appearing in the error terms is referred to as the *order* of the integration scheme. The total error in the calculation is the accumulation of these errors over all steps. Understanding these errors and how they scale with δt is critical to any study of numerical integration. Let's start by determining empirically how the errors in our calculation scale with δt.

Exercise 5.1: Run the program created in in-class Exercise 4.2(a) for several different values of δt — starting at 0.5 and decreasing in size by factors of 2 until $\delta t < 10^{-4}$ — and investigate how the energy error ΔE, defined as

$$\Delta E(\delta t) \;\equiv\; \max_i \left| E(t_i) - E(0) \right|,$$

depends on δt, where $t_i = i\delta t$ and the maximum is computed over all time steps. Specifically, compute the motion up to time $t_{max} = 20$, plot $\log_{10}\,\Delta E$ against $\log_{10}\,\delta t$ and determine the slope of the straight line that best fits the data. [Note that the C++/Python function returning $\log_{10}$ is called `log10` — the `log` function returns the natural logarithm (ln or $\log_e$) of its argument.]