

5.7: Numerical Approaches

If the amplitude of oscillations, for whatever reason, becomes so large that nonlinear terms in the equation describing an oscillator become comparable with its linear terms, numerical methods are virtually the only avenue available for their theoretical studies. In Hamiltonian 1D systems, such methods may be applied directly to Eq. (3.26), but dissipative and/or parametric systems typically lack such first integrals of motion, so that the initial differential equation has to be solved.

Let us discuss the general idea of such methods on the example of what mathematicians call the Cauchy problem (finding the solution for all moments of time, starting from the known initial conditions) for the first-order differential equation

$$\dot{q} = f(t, q). \quad (5.7.1)$$

(The generalization to a system of several such equations is straightforward.) Breaking the time axis into small, equal steps h (Figure 11) we can reduce the equation integration problem to finding the function's value at the next time point, $q_{n+1} \equiv q(t_{n+1}) \equiv q(t_n + h)$ from the previously found value $q_n = q(t_n)$ – and, if necessary, the values of q at other previous time steps.

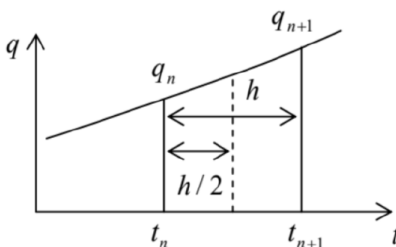


Figure 5.11. The basic notions used at numerical integration of ordinary differential equations.

In the simplest approach (called the Euler method), q_{n+1} is found using the following formula:

$$\begin{aligned} q_{n+1} &= q_n + k, \\ k &\equiv h f(t_n, q_n). \end{aligned}$$

This approximation is equivalent to the replacement of the genuine function $q(t)$, on the segment $[t_n, t_{n+1}]$, with the two first terms of its Taylor expansion in point t_n :

$$q(t_n + h) \approx q(t_n) + \dot{q}(t_n) h \equiv q(t_n) + h f(t_n, q_n). \quad (5.7.2)$$

This approximation has an error proportional to h^2 . One could argue that making the step h sufficiently small, the Euler method's error might be made arbitrarily small, but even with all the number-crunching power of modern computer platforms, the CPU time necessary to reach sufficient accuracy may be too large for large problems.³⁵ Besides that, the increase of the number of time steps, which is necessary at $h \rightarrow 0$ at a fixed total time interval, increases the total rounding errors and eventually may cause an increase, rather than the reduction of the overall error of the computed result.

A more efficient way is to modify Eq. (96) to include the terms of the second order in h . There are several ways to do this, for example using the 2nd-order Runge-Kutta method:

$$\begin{aligned} q_{n+1} &= q_n + k_2, \\ k_2 &\equiv h f\left(t_n + \frac{h}{2}, q_n + \frac{k_1}{2}\right), \quad k_1 \equiv h f(t_n, q_n). \end{aligned}$$

One can readily check that this method gives the exact result if the function $q(t)$ is a quadratic polynomial, and hence in the general case its errors are of the order of h^3 . We see that the main idea here is to first break the segment $[t_n, t_{n+1}]$ in half (see Figure 11), evaluate the right-hand side of the differential equation (95) at the point intermediate (in both t and q) between the points number n and $(n+1)$, and then use this information to predict q_{n+1} .

The advantage of the Runge-Kutta approach over other second-order methods is that it may be readily extended to the 4th order, without an additional breakup of the interval $[t_n, t_{n+1}]$:

$$q_{n+1} = q_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_4 \equiv hf(t_n + h, q_n + k_3), k_3 \equiv hf\left(t_n + \frac{h}{2}, q_n + \frac{k_2}{2}\right), k_2 \equiv hf\left(t_n + \frac{h}{2}, q_n + \frac{k_1}{2}\right), k_1 \equiv hf(t_n, q_n).$$

This method has a much lower error, $O(h^5)$, without being not too cumbersome. These features have made the 4th-order Runge-Kutta the default method in most numerical libraries. Its extension to higher orders is possible, but requires more complex formulas, and is justified only for some special cases, e.g., very abrupt functions $q(t)$.³⁶ The most frequent enhancement of the method is an automatic adjustment of the step h to reach the pre-specified accuracy, but not make more calculations than necessary.

Figure 12 shows a typical example of an application of that method to the very simple problem of a damped linear oscillator, for two values of fixed time step h (expressed in terms of the number N of such steps per oscillation period). The black straight lines connect the adjacent points obtained by the 4th-order Runge-Kutta method, while the points connected with the green straight lines represent the exact analytical solution (22). The plots show that a-few-percent errors start to appear only at as few as ~ 10 time steps per period, so that the method is indeed very efficient.

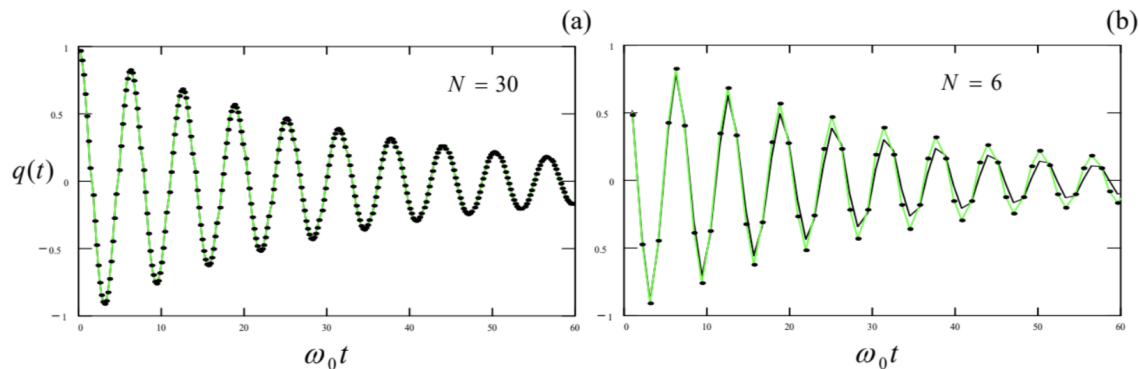


Figure 5.12. Results of the Runge-Kutta solution of Eq. (6) (with $\delta/\omega_0 = 0.03$) for: (a) 30 and (b) 6 points per oscillation period. The results are shown by points; the black and green lines are only the guides for the eye.

Let me hope that the discussion in the next section will make the conveniences and the handicaps of the numerical approach to problems of nonlinear dynamics very clear.

³⁵ In addition, the Euler method is not time-reversible - the handicap that may be essential for the integration of Hamiltonian systems described by systems of second-order differential equations. However, this drawback may be partly overcome by the so-called leapfrogging - the overlap of time steps h for a generalized coordinate and the corresponding generalized velocity.

³⁶ The most popular approaches in such cases are the Richardson extrapolation, the Bulirsch-Stoer algorithm, and a set of so-called prediction-correction techniques, e.g. the Adams-Bashforth-Moulton method - see the literature recommended in MA Sec. 16(iii).