sine of $\sqrt{10\pi}t$, is generated by the initial conditions $\phi_1(0) = 0$, $\phi_2(0) = \sqrt{10\pi}$. While analytically we can tell the difference between $\cosh \sqrt{10\pi}t$ and $\sinh \sqrt{10\pi}t$, for large $t$ we have $\cosh \sqrt{10\pi}t \sim e^{\sqrt{10\pi}t}/2$ and $\sinh \sqrt{10\pi}t \sim e^{\sqrt{10\pi}t}/2$; numerically these two functions look exactly the same if only a fixed number of digits are retained. For example, correct to eight significant figures, we find that for $t = 1$

$$\sinh \sqrt{10\pi} = \cosh \sqrt{10\pi} = 10,315.894.$$ 

If the calculations are carried out on a machine that carries only eight digits, the two solutions $\phi_1$ and $\phi_2$ are identical at $t = 1$ and indeed for all $t > 1$. Thus, even though the solutions are linearly independent, their numerical tabulation would show that they are the same because we can retain only a finite number of digits. This phenomenon is called numerical dependence.

For the present problem we can partially circumvent this difficulty by computing, instead of $\sinh \sqrt{10\pi}t$ and $\cosh \sqrt{10\pi}t$, the linearly independent solutions $\phi_3(t) = e^{\sqrt{10\pi}t}$ and $\phi_4(t) = e^{-\sqrt{10\pi}t}$ corresponding to the initial conditions $\phi_3(0) = 1$, $\phi_4(0) = \sqrt{10\pi}$ and $\phi_3(0) = 1$, $\phi_4(0) = -\sqrt{10\pi}$, respectively. The solution $\phi_3$ grows exponentially while $\phi_4$ decays exponentially. Even so, we encounter difficulty in calculating $\phi_4$ correctly on a large interval. The reason is that at each step of the calculation for $\phi_4$ we introduce truncation and round-off errors. Thus at any point $t_n$ the data to be used in going to the next point are not precisely the values of $\phi_4(t_n)$ and $\phi_4(t_n)$. The solution of the initial value problem with these data at $t_n$ involves not only $e^{-\sqrt{10\pi}t}$ but also $e^{\sqrt{10\pi}t}$. Because the error in the data at $t_n$ is small, the latter function appears with a very small coefficient. Nevertheless, since $e^{-\sqrt{10\pi}t}$ tends to zero and $e^{\sqrt{10\pi}t}$ grows very rapidly, the latter eventually dominates, and the calculated solution is simply a multiple of $e^{\sqrt{10\pi}t} = \phi_4(t)$.

To be specific, suppose that we use the Runge–Kutta method to calculate the solution $y = \phi_4(t) = e^{-\sqrt{10\pi}t}$ of the initial value problem

$$y'' - 10\pi^2 y = 0, \quad y(0) = 1, \quad y'(0) = -\sqrt{10\pi}.$$  

(The Runge–Kutta method for second order systems is described in Section 8.6.) Using single-precision (eight-digit) arithmetic with a step size $h = 0.01$, we obtain the results in Table 8.5.4. It is clearly evident from these results that the numerical solution begins to deviate significantly from the exact solution for $t > 0.5$, and soon differs from it by many orders of magnitude. The reason is the presence in the numerical solution of a small component of the exponentially growing solution $\phi_3(t) = e^{\sqrt{10\pi}t}$. With eight-digit arithmetic we can expect a round-off error of the order of $10^{-8}$ at each step. Since $e^{\sqrt{10\pi}t}$ grows by a factor of $3.7 \times 10^{21}$ from $t = 0$ to $t = 5$, an error of order $10^{-8}$ near $t = 0$ can produce an error of order $10^{13}$ at $t = 5$ even if no further errors are introduced in the intervening calculations. The results given in Table 8.5.4 demonstrate that this is exactly what happens.

Equation (18) is highly unstable and the behavior shown in this example is typical of unstable problems. One can track a solution accurately for a while, and the interval can be extended by using smaller step sizes or more accurate methods, but eventually the instability in the problem itself takes over and leads to large errors.