Comments on numerical methods

Custom Fortran codes were used for obtaining numerical solutions of ordinary and stochastic differential equations.

For ordinary differential equations, the following initial value problem was considered:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}, t) && (1) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 && (2) \end{aligned}$$

Here $\mathbf{f} = (f_1, f_2, \ldots, f_j, \ldots, f_N)$ is a vector function of a state variables vector $\mathbf{x} = (x_1, x_2, \ldots, x_j, \ldots x_N)$ and time t. A numerical solution is obtained on a uniform grid: $t_{k+1} = t_k + h$; h is the integration step size. So, the numerical solution provides the state variables at grid nodes: $\mathbf{x}(t_k) = \mathbf{x}_k$.

Runge-Kutta methods are a one-step scheme that provides the solution at node $k+1$ using the known values of the state variables at previous node k.

Consider a differential equation for a component f_j of the vector function $\mathbf{f}$:

$$\frac{dx_j}{dt} = f_j(\mathbf{x}, t) \quad (3)$$

For Eq. (3), one step of the numerical methods is the following
Euler method:

$$x_{j,k+1} = x_{j,k} + hf_j(\mathbf{x}_k, t_k) \quad (4)$$

classical 4th order Runge-Kutta method:

$$\begin{aligned} G_{j,1} &= hf_j(\mathbf{x}_k, t_k) \\ G_{j,2} &= hf_j(\mathbf{x}_k + 0.5G_{j,1}, t_k + 0.5h) \\ G_{j,3} &= hf_j(\mathbf{x}_k + 0.5G_{j,2}, t_k + 0.5h) \\ G_{j,4} &= hf_j(\mathbf{x}_k + G_{j,3}, t_k + h) \\ x_{j,k+1} &= x_{j,k} + \left[G_{j,1} + 2G_{j,2} + 2G_{j,3} + G_{j,4}\right]/6 && (5) \end{aligned}$$

Ralston method:

$$\begin{aligned} G_{j,1} &= hf_j(\mathbf{x}_k, t_k) \\ G_{j,2} &= hf_j(\mathbf{x}_k + 0.4G_{j,1}, t_k + 0.4h) \\ G_{j,3} &= hf_j(\mathbf{x}_k + s_1G_{j,1} + s_2Gj, 2, t_k + c_1h) \\ G_{j,4} &= hf_j(\mathbf{x}_k + s_3G_{j,1} + s_4G_{j,2} + s_5G_{j,3}, t_k + h) \\ x_{j,k+1} &= x_{j,k} + \left[s_6G_{j,1} + s_7G_{j,2} + s_8G_{j,3} + s_9G_{j,4}\right] && (6) \end{aligned}$$

where constants in computer double precision format are $c_1 = 0.45573725421878941D0$, $s_1 = 0.29697760924775363D0$, $s_2 = 0.15875964497103556D0$, $s_3 = 0.21810038822592048D0$, $s_4 = -0.3050965148692931D1$, $s_5 = 0.38328647604670101D1$, $s_6 = 0.17476028226269039D0$, $s_7 = -0.55148066287873299D0$, $s_8 = 0.12055355993965235D1$, $s_9 = 0.17118478121951902D0$.

The material on the web page http://www.mymathlib.com/diffeq/runge-kutta/runge_kutta_ralston_4.html was used for calculating the constants.

For stochastic differential equations, the following initial value problem was considered:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}, t) + \sqrt{D}\xi(t) && (7) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 && (8) \end{aligned}$$

Consider a stochastic differential equation for a component f_j of the vector function $\mathbf{f}$:

$$\frac{dx_j}{dt} = f_j(\mathbf{x}, t) + \sqrt{D}\xi_j(t) \tag{9}$$

For Eq. (9), one step of **the stochastic 4th Runge-Kutta numerical method** is the following

$$\begin{aligned}
W_{j,k} &= \sqrt{Dh}\,\mathrm{grnd}(1) \\
G_{j,1} &= hf_j(\mathbf{x}_k, t_k) + Wj, k \\
G_{j,2} &= hf_j(\mathbf{x}_k + 0.5G_{j,1}, t_k + 0.5h) + Wj, k \\
G_{j,3} &= hf_j(\mathbf{x}_k + 0.5G_{j,2}, t_k + 0.5h) + Wj, k \\
G_{j,4} &= hf_j(\mathbf{x}_k + G_{j,3}, t_k + h) + Wj, k \\
x_{j,k+1} &= x_{j,k} + \left[G_{j,1} + 2G_{j,2} + 2G_{j,3} + G_{j,4}\right]/6 \qquad (10)
\end{aligned}$$

where $\mathtt{grnd}(1)$ is a generator of independent, normally distributed random numbers with zero mean and unit variance. A custom code using Marsaglia's KISS algorithm (www.jstatsoft.org/v08/i14/paper) and the Box-Muller transformation (https://www.jstor.org/stable/2237361) was used.