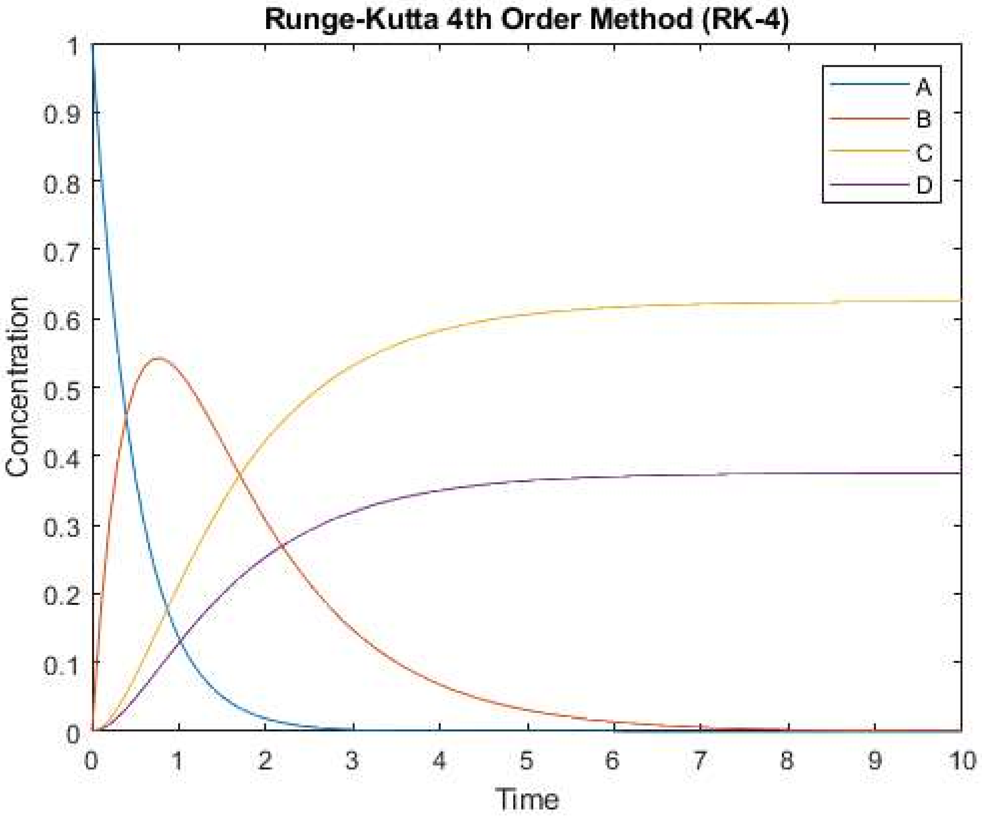
```
t0 = 0;         % Initial time
tEnd = 10;      % Final time
h = 0.1;        % Step size
N = (tEnd - t0) / h;        % Number of data points
T = t0:h:tEnd;              % Time vector
k1 = 2;         % Reaction constant for A to B
k2 = 0.5;       % Reaction constant for B to C
k3 = 0.3;       % Reaction constant for B to D
Ao = 1;         % Initial value of A at t = 0
Bo = 0;         % Initial value of B at t = 0
Co = 0;         % Initial value of C at t = 0
Do = 0;         % Initial value of D at t = 0
A = zeros(N+1, 1);  % Vector to store values of A
B = zeros(N+1, 1);  % Vector to store values of B
C = zeros(N+1, 1);  % Vector to store values of C
D = zeros(N+1, 1);  % Vector to store values of D
A(1) = Ao;
B(1) = Bo;
C(1) = Co;
D(1) = Do;

for i = 1:N  % Calculating intermediate slopes for A,B,C,D
    t = T(i);

    k1a = f1(A(i), B(i), C(i), D(i), t);
    k1b = f2(A(i), B(i), C(i), D(i), t);
    k1c = f3(A(i), B(i), C(i), D(i), t);
    k1d = f4(A(i), B(i), C(i), D(i), t);

    k2a = f1(A(i) + 0.5*h*k1a, B(i) + 0.5*h*k1b, C(i) + 0.5*h*k1c, D(i) +
 0.5*h*k1d, t + 0.5*h);
    k2b = f2(A(i) + 0.5*h*k1a, B(i) + 0.5*h*k1b, C(i) + 0.5*h*k1c, D(i) +
 0.5*h*k1d, t + 0.5*h);
    k2c = f3(A(i) + 0.5*h*k1a, B(i) + 0.5*h*k1b, C(i) + 0.5*h*k1c, D(i) +
 0.5*h*k1d, t + 0.5*h);
    k2d = f4(A(i) + 0.5*h*k1a, B(i) + 0.5*h*k1b, C(i) + 0.5*h*k1c, D(i) +
 0.5*h*k1d, t + 0.5*h);

    k3a = f1(A(i) + 0.5*h*k2a, B(i) + 0.5*h*k2b, C(i) + 0.5*h*k2c, D(i) +
 0.5*h*k2d, t + 0.5*h);
    k3b = f2(A(i) + 0.5*h*k2a, B(i) + 0.5*h*k2b, C(i) + 0.5*h*k2c, D(i) +
 0.5*h*k2d, t + 0.5*h);
    k3c = f3(A(i) + 0.5*h*k2a, B(i) + 0.5*h*k2b, C(i) + 0.5*h*k2c, D(i) +
 0.5*h*k2d, t + 0.5*h);
    k3d = f4(A(i) + 0.5*h*k2a, B(i) + 0.5*h*k2b, C(i) + 0.5*h*k2c, D(i) +
 0.5*h*k2d, t + 0.5*h);

    k4a = f1(A(i) + h*k3a, B(i) + h*k3b, C(i) + h*k3c, D(i) + h*k3d, t + h);
    k4b = f2(A(i) + h*k3a, B(i) + h*k3b, C(i) + h*k3c, D(i) + h*k3d, t + h);
    k4c = f3(A(i) + h*k3a, B(i) + h*k3b, C(i) + h*k3c, D(i) + h*k3d, t + h);
    k4d = f4(A(i) + h*k3a, B(i) + h*k3b, C(i) + h*k3c, D(i) + h*k3d, t + h);
    % Applying RK-4 method
    A(i+1) = A(i) + (h/6) * (k1a + 2*k2a + 2*k3a + k4a);
    B(i+1) = B(i) + (h/6) * (k1b + 2*k2b + 2*k3b + k4b);
    C(i+1) = C(i) + (h/6) * (k1c + 2*k2c + 2*k3c + k4c);
    D(i+1) = D(i) + (h/6) * (k1d + 2*k2d + 2*k3d + k4d);
end
plot(T, A) % plotting A vs T curve
hold on
plot(T, B) % plotting B vs T curve
plot(T, C) % plotting C vs T curve
plot(T, D) % plotting D vs T curve
legend('A','B','C','D');
xlabel('Time');
ylabel('Concentration');
title('Runge-Kutta 4th Order Method (RK-4)');
function dAdt = f1(A, B, C, D, t) % function for calculating dA/dT
    k1 = 2;
    dAdt = -k1 * A;
end
function dBdt = f2(A, B, C, D, t) % function for calculating dB/dT
    k1 = 2;
    k2 = 0.5;
    k3 = 0.3;
    dBdt = k1 * A - k2 * B - k3 * B;
end
function dCdt = f3(A, B, C, D, t) % function for calculating dC/dT
    k2 = 0.5;
    dCdt = k2 * B;
end
function dDdt = f4(A, B, C, D, t) % function for calculating dD/dT
    k3 = 0.3;
    dDdt = k3 * B;
end
```



Published with MATLAB® R2022b