Prof. P. Koumoutsakos, G. Tauriello
ETH Zentrum, CLT F 12, C 11
CH-8092 Zürich

Solution 3

Issued: 1.4.2014

Question 1: Low-storage Runge-Kutta time-stepping

For the considered ODE

$$\frac{dy}{dt} = f(y(t)), \tag{1}$$

with a given initial condition $y(t = 0)$, the third order low storage Runge-Kutta (LS-RK3) scheme is given by

$$\begin{cases} q_j = a_j q_{j-1} + \delta t \ f(y_{j-1}), \\ y_j = y_{j-1} + b_j q_j, \end{cases} \qquad j = 1 \dots 3, \tag{2}$$

where δt is the time-step. At every n-th time-step, we set $y_0 = y^n$ and obtain $y^{n+1} = y_3$ where $y^n \approx y(t = n\delta t)$.

1. First step:

$$q_1 = \delta t f(y_0) \ , \tag{3}$$

$$y_1 = y_0 + \frac{1}{4}\delta t f(y_0) \ . \tag{4}$$

Second step:

$$q_2 = -\frac{17}{32}\delta t f(y_0) + \delta t f\left(y_0 + \frac{1}{4}\delta t f(y_0)\right) \ , \tag{5}$$

$$y_2 = y_0 - \frac{2}{9}\delta t f(y_0) + \frac{8}{9}\delta t f\left(y_0 + \frac{1}{4}\delta t f(y_0)\right) \ . \tag{6}$$

Third step:

$$\begin{aligned} q_3 = &\frac{17}{27}\delta t f(y_0) - \frac{32}{27}\delta t f\left(y_0 + \frac{1}{4}\delta t f(y_0)\right) + \\ &+ \ \delta t f\left(y_0 - \frac{2}{9}\delta t f(y_0) + \frac{8}{9}\delta t f\left(y_0 + \frac{1}{4}\delta t f(y_0)\right)\right) \ , \end{aligned} \tag{7}$$

$$y_3 = y_0 + \frac{1}{4}\delta t f(y_0) + \frac{3}{4}\delta t f\left(y_0 - \frac{2}{9}\delta t f(y_0) + \frac{8}{9}\delta t f\left(y_0 + \frac{1}{4}\delta t f(y_0)\right)\right) \ . \tag{8}$$

Finally, since $y_0 = y^n$ and $y^{n+1} = y_3$, we obtain:

$$y^{n+1} = y^n + \frac{1}{4}\delta t f(y^n) + \frac{3}{4}\delta t f\left(y^n - \frac{2}{9}\delta t f(y^n) + \frac{8}{9}\delta t f\left(y^n + \frac{1}{4}\delta t f(y^n)\right)\right) \ . \tag{9}$$

For $f(y) = y$:

$$y^{n+1} = y^n + \frac{1}{4}\delta t y^n + \frac{3}{4}\delta t\left(y^n - \frac{2}{9}\delta t y^n + \frac{8}{9}\delta t\left(y^n + \frac{1}{4}\delta t y^n\right)\right) \ . \tag{10}$$

Rearranging, the equation looks like:

$$y^{n+1} = y^n + \delta t y^n + \frac{1}{2}\delta t^2 y^n + \frac{1}{6}\delta t^3 y^n \ . \tag{11}$$

2. The exact solution of Eq. 1 with $f(y(t)) = y(t)$ is $y(t) = y^0\, e^t$ with $y^0 = y(t = 0)$ and the expression in Eq. 11 is the Taylor expansion up to the third order of $y(t)$ at $t = n\delta t$. It is consistent with the expected order of the method which has a local error of $\mathcal{O}(\delta t^4)$ at each time-step and thus a global error of $\mathcal{O}(\delta t^3)$ for any given end time T.

Question 2: Phase Space Diagrams - Harmonic Oscillator

```cpp
#include <iostream>
#include <fstream>
using namespace std;

class Particle
{
    double x, u, dxdt, dudt;

    void _rhs()
    {
        dxdt = u;
        dudt = -x;
    }

public:
    Particle(double x, double u): x(x), u(u), dxdt(0), dudt(0) {}

    void euler(double dt)
    {
        _rhs();

        x += dt*dxdt;
        u += dt*dudt;
    }

    void lsrk3(double dt)
    {
        // coefficients for LS-RK3
        const double a[3] = {0, -17./32, -32./27};
        const double b[3] = {1./4, 8./9, 3./4};

        // execute 3 stages of LS-RK3
        double q1 = 0, q2 = 0;
```

```
        for(int i = 0; i < 3; ++i)
        {
            _rhs();

            q1 = a[i]*q1 + dt*dxdt;
            q2 = a[i]*q2 + dt*dudt;

            x += b[i]*q1;
            u += b[i]*q2;
        }
    }

    double getx() const { return x; }
    double getu() const { return u; }
};

int main()
{
    ofstream out_euler("euler.dat");
    ofstream out_lsrk3("low-storage-runge-kutta3.dat");

    Particle p1(1, 0), p2(1, 0);

    const double dt = 5e-2;

    const int N = 20/dt;
    for(int i=0; i<N; i++)
    {
        out_euler << p1.getx() << "\t" << p1.getu() << "\n";
        out_lsrk3 << p2.getx() << "\t" << p2.getu() << "\n";

        p1.euler(dt);
        p2.lsrk3(dt);
    }

    return 0;
}
```

Listing 1: main.cpp

Figure 1 shows the phase space diagram obtained using Forward Euler and LS-RK3 schemes with $\delta t = 0.05$. Using Euler, the trajectory diverges whereas with LS-RK3 it does not.

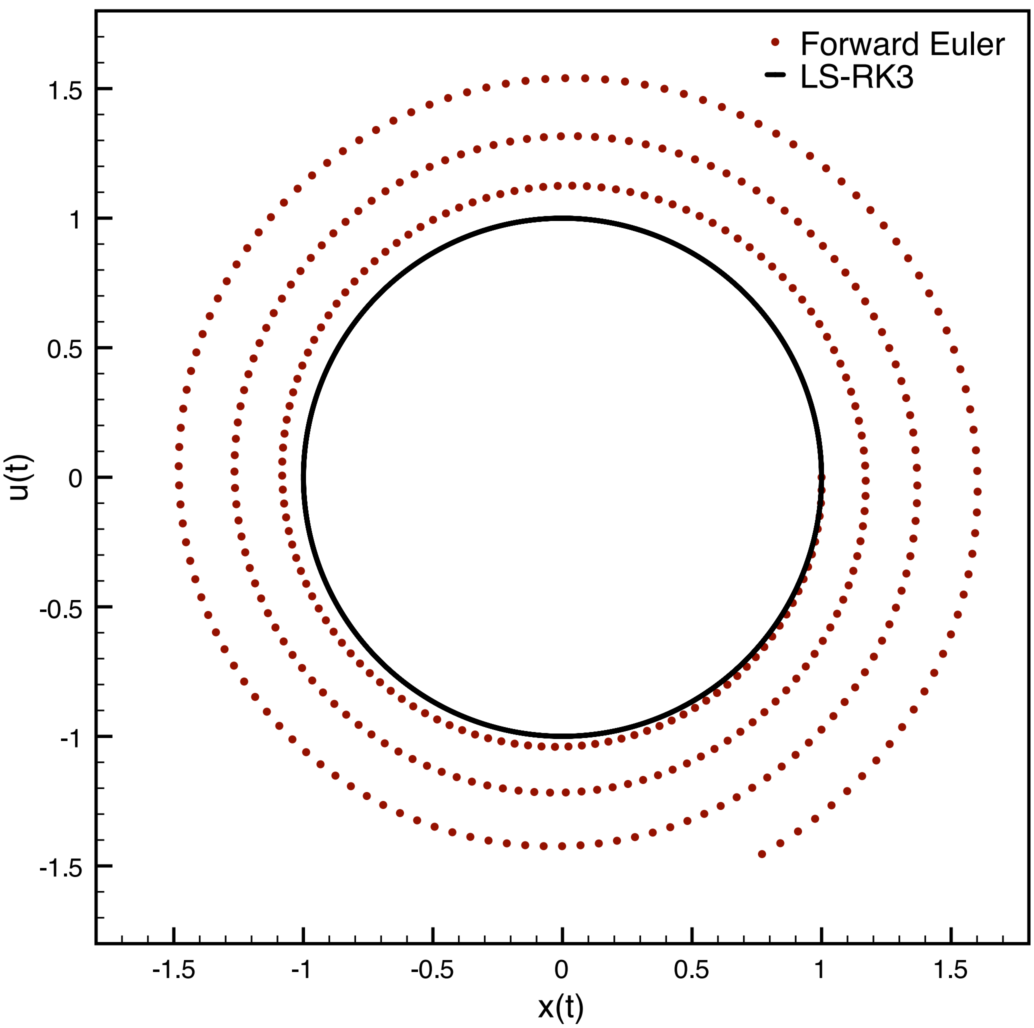

Figure 1: Phase space diagram