NUM2cv
24. dubna 2017
Vladislav Belov

1 Rungeovy-Kuttovy metody

Řešíme numericky Riccatiho rovnici (1) na intervalu $[0.25, 0.45]$ Rungeovými-Kuttovými metodami. Známe její analytické řešení: $u(t) = (\frac{1}{\sqrt{2}t}\tan(\sqrt{2}(c - \frac{1}{t})) - \frac{1}{2t})e^t$, kde klademe $c = 1$.

$$\begin{aligned} \dot{u}(t) &= t^{-4}e^t + u(t) + 2e^{-t}u^2(t) = f(t,u), \\ u(0.25) &= -31,1844. \end{aligned} \tag{1}$$

Označíme $\tau = integrationTimeStep$, $u_0 = u(t_0)$. Použijeme metody:

I. Runge-Kutta 1. řádu (Euler):

$$\begin{aligned} k_1(\tau) &= \tau \cdot f(t_0, u_0), \\ u(t_0 + \tau) &= u(t) + k_1(\tau). \end{aligned}$$

II. Runge-Kutta 2. řádu:

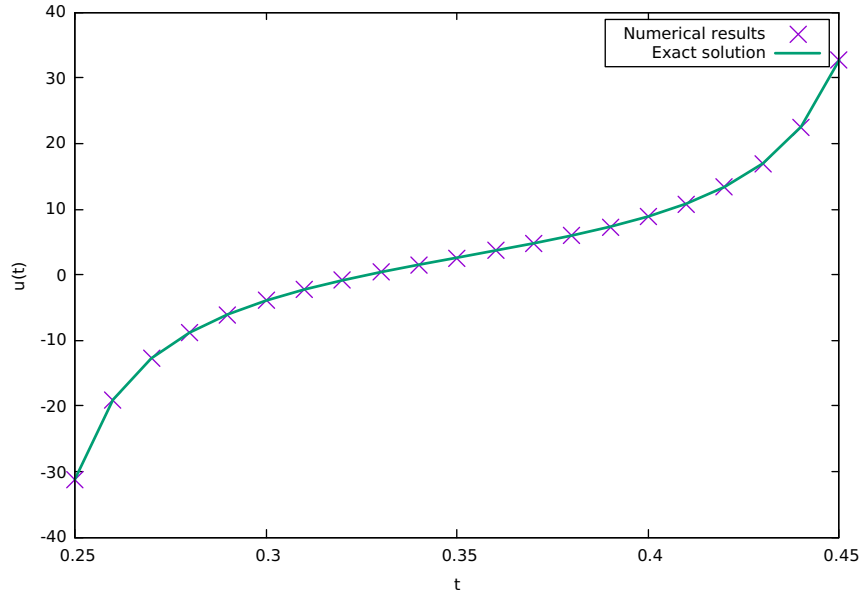
$$\begin{aligned} k_1(\tau) &= \tau \cdot f(t_0, u_0), \\ k_2(\tau) &= \tau \cdot f(t_0 + \tau, u_0 + k_1(\tau)), \\ u(t_0 + \tau) &= u(t) + \tfrac{1}{2}k_1(\tau) + \tfrac{1}{2}k_2(\tau). \end{aligned}$$

III. Runge–Kutta–Merson:

$$\begin{aligned} k_1(\tau) &= \tau \cdot f(t_0, u_0), \\ k_2(\tau) &= \tau \cdot f(t_0 + \tfrac{1}{3}\tau, u_0 + \tfrac{1}{3}k_1), \\ k_3(\tau) &= \tau \cdot f(t_0 + \tfrac{1}{3}\tau, u_0 + \tfrac{1}{6}k_1 + \tfrac{1}{6}k_2), \\ k_4(\tau) &= \tau \cdot f(t_0 + \tfrac{1}{2}\tau, u_0 + \tfrac{1}{8}k_1 + \tfrac{3}{8}k_3), \\ k_5(\tau) &= \tau \cdot f(t_0 + \tau, u_0 + \tfrac{1}{2}k_1 - \tfrac{3}{2}k_3 + 2k_4), \\ u(t_0 + \tau) &= y_0 + \tfrac{1}{6}k_1 + \tfrac{2}{3}k_4 + \tfrac{1}{6}k_5. \end{aligned}$$

Soustředíme se na metodě (II):

(a) Průběh numerického řešení v porovnání s průběhem analytického ($timeStep = 10^{-2}$, $integrationTimeStep = 10^{-3}$):



(b) Implementace:

```
template< typename Problem >
class RungeKutta : public IntegratorBase
{
 public:

  RungeKutta( Problem& problem )
  {
        this->k1 = new double[ problem.getDegreesOfFreedom() ];
        this->k2 = new double[ problem.getDegreesOfFreedom() ];
        this->aux = new double[ problem.getDegreesOfFreedom() ];
  }

  bool solve( Problem& problem,
              double* u )
  {
        const int dofs = problem.getDegreesOfFreedom();
        double tau = std::min( this->integrationTimeStep, this->stopTime - this->time );
        long int iteration( 0 );
        while( this->time < this->stopTime )
        {
          /***
          * Compute k1
          */
          problem.getRightHandSide( this->time, u, k1 );

          /***
          * Compute k2
          */
          for( int i = 0; i < dofs; i++ )
            aux[ i ] = u[ i ] + tau * k1[ i ];
          problem.getRightHandSide( this->time + tau, aux, k2 );

          /****/
          for( int i = 0; i < dofs; i++ )
            u[ i ] += ( tau / 2.0 ) * ( k1[ i ] + k2[ i ] );
          this->time += tau;
          iteration++;
          if( iteration > 100000 )
          {
            std::cerr << "The solver has reached the maximum number of iteratoins. "
                      << std::endl;
            return false;
          }
          tau = std::min( tau, this->stopTime - this->time );
          std::cout << "ITER: " << iteration << " \t tau = " << tau
                      << " \t time= " << time << "          \r " << std::flush;
        }
        std::cout << std::endl;
        return true;
  }

  ~RungeKutta()
  {
        delete[] k1;
        delete[] k2;
        delete[] aux;
  }

 protected:

  double *k1, *k2, *aux;

};
```

2 L^1, L^2, L^∞ - normy, EOC (experimental orders of convergence)

Označíme $\tau = integrationTimeStep$, $\Delta t = timeStep$ (pro výpočet norem položíme $\tau = \Delta t$), $\bar{u}_\tau$ - numerické řešení spočítané s integračním krokem τ, u - analytické řešení, K - počet bodů v rozdělení intervalu $[a, b]$ ($a = 0.25$, $b = 0.45$, $K = \frac{b-a}{\Delta t}$). Potom lze spočítat normy takto:

$$\|\bar{u}_\tau - u\|_{L^1} = \sum_{j=0}^{K} |\bar{u}_\tau(a + j\Delta t) - u(a + j\Delta t)| \cdot \Delta t$$

$$\|\bar{u}_\tau - u\|_{L^2} = \Big(\sum_{j=0}^{K} |\bar{u}_\tau(a + j\Delta t) - u(a + j\Delta t)|^2 \cdot \Delta t \Big)^{\frac{1}{2}}$$

$$\|\bar{u}_\tau - u\|_{L^\infty} = \max_{j=0,1,\cdots,K} |\bar{u}_\tau(a + j\Delta t) - u(a + j\Delta t)|$$

Chyba (err.) metody se spočte jako $E_\tau = \|\bar{u}_\tau - u\|_L$ a experimentální řád konvergence $\mathrm{EOC}(E_{\tau_1}, E_{\tau_2}) = \log_2 \frac{E_{\tau_1}}{E_{\tau_2}} / \log_2 \frac{\tau_1}{\tau_2}$.

τ	L^1		L^2		L^∞	
	err.	EOC	err.	EOC	err.	EOC
$2 \cdot 10^{-3}$	$3.27189 \cdot 10^{-1}$	×	1.18346	×	11.7741	×
$1 \cdot 10^{-3}$	$1.49554 \cdot 10^{-1}$	1.12946	$5.22467 \cdot 10^{-1}$	1.1796	5.20364	1.17802
$5 \cdot 10^{-4}$	$7.17241 \cdot 10^{-2}$	1.06013	$2.46576 \cdot 10^{-1}$	1.0833	2.45653	1.0829
$2.5 \cdot 10^{-4}$	$3.51473 \cdot 10^{-2}$	1.02904	$1.19901 \cdot 10^{-1}$	1.04019	1.19462	1.25532
$1.25 \cdot 10^{-4}$	$1.74006 \cdot 10^{-2}$	1.01428	$5.91353 \cdot 10^{-2}$	1.01975	$5.89204 \cdot 10^{-1}$	0.80446
$6.25 \cdot 10^{-5}$	$8.65769 \cdot 10^{-3}$	1.00708	$2.93677 \cdot 10^{-2}$	1.00979	$2.92613 \cdot 10^{-1}$	1.00977

Tabulka 1: Eulerova metoda

τ	L^1		L^2		L^∞	
	err.	EOC	err.	EOC	err.	EOC
$2 \cdot 10^{-3}$	$6.09946 \cdot 10^{-3}$	×	$2.39973 \cdot 10^{-2}$	×	$2.70752 \cdot 10^{-1}$	×
$1 \cdot 10^{-3}$	$1.48308 \cdot 10^{-3}$	2.04009	$5.79622 \cdot 10^{-3}$	2.04969	$6.82516 \cdot 10^{-2}$	1.98804
$5 \cdot 10^{-4}$	$3.64603 \cdot 10^{-4}$	2.0242	$1.41923 \cdot 10^{-3}$	2.03	$1.70738 \cdot 10^{-2}$	1.99908
$2.5 \cdot 10^{-4}$	$9.03309 \cdot 10^{-5}$	2.01303	$3.50848 \cdot 10^{-4}$	2.01619	$4.26622 \cdot 10^{-3}$	2.00075
$1.25 \cdot 10^{-4}$	$2.24775 \cdot 10^{-5}$	2.00674	$8.72043 \cdot 10^{-5}$	2.00837	$1.06606 \cdot 10^{-3}$	2.00067
$6.25 \cdot 10^{-5}$	$5.60607 \cdot 10^{-6}$	2.00342	$2.17368 \cdot 10^{-5}$	2.00426	$2.66439 \cdot 10^{-4}$	2.00041

Tabulka 2: Rungeova-Kuttova metoda 2. řádu

τ	L^1		L^2		L^∞	
	err.	EOC	err.	EOC	err.	EOC
$2 \cdot 10^{-3}$	$3.2243 \cdot 10^{-6}$	×	$1.21815 \cdot 10^{-5}$	×	$1.35241 \cdot 10^{-4}$	×
$1 \cdot 10^{-3}$	$1.94926 \cdot 10^{-7}$	4.04799	$7.27902 \cdot 10^{-7}$	4.0648	$8.3966 \cdot 10^{-6}$	4.00958
$5 \cdot 10^{-4}$	$1.19626 \cdot 10^{-8}$	4.02632	$4.44086 \cdot 10^{-8}$	4.03483	$5.22349 \cdot 10^{-7}$	4.00672
$2.5 \cdot 10^{-4}$	$7.40577 \cdot 10^{-10}$	4.01374	$2.74112 \cdot 10^{-9}$	4.018	$3.25602 \cdot 10^{-8}$	4.00383
$1.25 \cdot 10^{-4}$	$4.60616 \cdot 10^{-11}$	4.00701	$1.70236 \cdot 10^{-10}$	4.00916	$2.0321 \cdot 10^{-9}$	4.00207
$6.25 \cdot 10^{-5}$	$6.64083 \cdot 10^{-12}$	2.79413	$2.64732 \cdot 10^{-11}$	2.68493	$3.09228 \cdot 10^{-10}$	2.71623

Tabulka 3: Rungeova-Kuttova-Mersonova metoda (bez adaptivity)