NUM2cv
Domácí úkol č.2
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. 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:

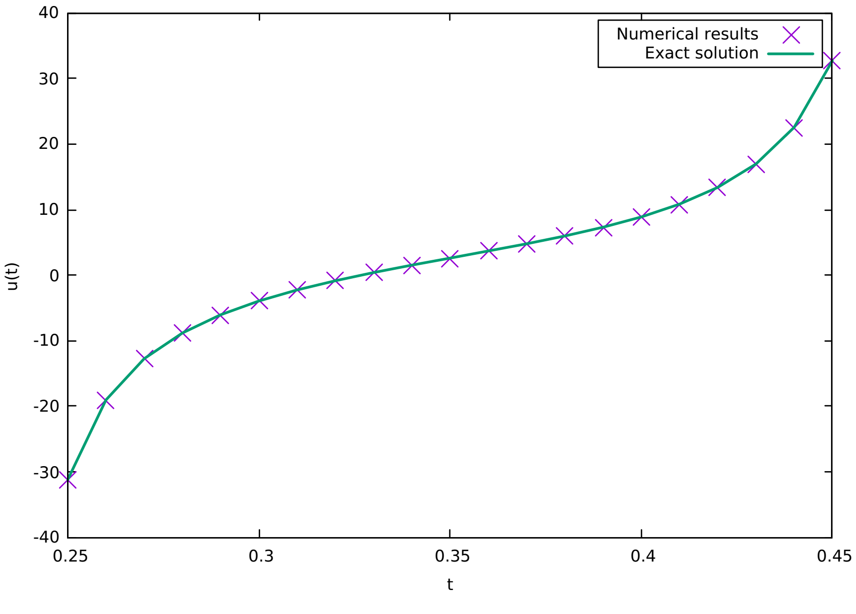
$$\begin{aligned} k_1(\tau) &= \tau \cdot f(t_0, u_0), \\ k_2(\tau) &= \tau \cdot f(t_0+\tau, u_0 + k_1(\tau)), \\ u(t_0+\tau) &= u(t) + \tfrac{1}{2}k_1(\tau) + \tfrac{1}{2}k_2(\tau). \end{aligned}$$

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 $\tau = integrationTimeStep$, $\Delta t = timeStep$ ($\Delta t = 10^{-2}$), $\bar{u}_\tau$ - numerické řešení spočítané s integračním krokem τ, u - analytické řešení, K - počet bodů v rozdělení intervalu $[a,b]$ ($a = 0.25$, $b = 0.45$, $K = \frac{b-a}{\Delta t} = \frac{0.45-0.25}{10^{-2}} = 20$). 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
10^{-3}	$1.75358 \cdot 10^{-1}$	$\times$	$6.52637 \cdot 10^{-1}$	$\times$	5.20364	$\times$
$5 \cdot 10^{-4}$	$8.44912 \cdot 10^{-2}$	1.053	$3.10358 \cdot 10^{-1}$	1.072	2.45653	1.083
$2.5 \cdot 10^{-4}$	$4.14977 \cdot 10^{-2}$	1.026	$1.51472 \cdot 10^{-1}$	1.035	1.19462	1.04
$1.25 \cdot 10^{-4}$	$2.05675 \cdot 10^{-2}$	1.013	$7.48415 \cdot 10^{-2}$	1.017	$5.89204 \cdot 10^{-1}$	1.02
$6.25 \cdot 10^{-5}$	$1.02391 \cdot 10^{-2}$	1.006	$3.72011 \cdot 10^{-2}$	1.008	$2.92613 \cdot 10^{-1}$	1.01
$3.125 \cdot 10^{-5}$	$5.10848 \cdot 10^{-3}$	1.003	$1.85461 \cdot 10^{-2}$	1.004	$1.45814 \cdot 10^{-1}$	1.005

Tabulka 1: Eulerova metoda

τ	L^1		L^2		L^∞	
	err.	EOC	err.	EOC	err.	EOC
10^{-3}	$1.83033 \cdot 10^{-3}$	$\times$	$7.84699 \cdot 10^{-3}$	$\times$	$6.82516 \cdot 10^{-2}$	$\times$
$5 \cdot 10^{-4}$	$4.55945 \cdot 10^{-4}$	2.0052	$1.96057 \cdot 10^{-3}$	2.00087	$1.70738 \cdot 10^{-2}$	1.99908
$2.5 \cdot 10^{-4}$	$1.13708 \cdot 10^{-4}$	2.0035	$4.89612 \cdot 10^{-4}$	2.00156	$4.26622 \cdot 10^{-3}$	2.00075
$1.25 \cdot 10^{-4}$	$2.83878 \cdot 10^{-5}$	2.002	$1.22314 \cdot 10^{-4}$	2.00105	$1.06606 \cdot 10^{-3}$	2.00067
$6.25 \cdot 10^{-5}$	$7.0918 \cdot 10^{-6}$	2.001	$3.05658 \cdot 10^{-5}$	2.0006	$2.66439 \cdot 10^{-4}$	2.00041
$3.125 \cdot 10^{-5}$	$1.77228 \cdot 10^{-6}$	2.0005	$7.63974 \cdot 10^{-6}$	2.00032	$6.65992 \cdot 10^{-5}$	2.00023

Tabulka 2: Rungeova-Kuttova metoda 2. řádu

τ	L^1		L^2		L^∞	
	err.	EOC	err.	EOC	err.	EOC
10^{-3}	$2.36651 \cdot 10^{-7}$	$\times$	$9.76783 \cdot 10^{-7}$	$\times$	$8.3966 \cdot 10^{-6}$	$\times$
$5 \cdot 10^{-4}$	$1.46934 \cdot 10^{-8}$	4.00952	$6.07233 \cdot 10^{-8}$	4.00772	$5.22349 \cdot 10^{-7}$	4.00672
$2.5 \cdot 10^{-4}$	$9.1339 \cdot 10^{-10}$	4.00779	$3.77666 \cdot 10^{-9}$	4.00707	$3.24991 \cdot 10^{-8}$	4.00654
$1.25 \cdot 10^{-4}$	$5.54893 \cdot 10^{-11}$	4.04095	$2.28997 \cdot 10^{-10}$	4.04371	$1.97127 \cdot 10^{-9}$	4.0432
$6.25 \cdot 10^{-5}$	$8.35626 \cdot 10^{-12}$	2.73128	$3.62685 \cdot 10^{-11}$	2.65854	$3.10102 \cdot 10^{-10}$	2.66831
$3.125 \cdot 10^{-5}$	$7.7862 \cdot 10^{-12}$	0.10194	$3.51773 \cdot 10^{-11}$	0.04407	$2.98606 \cdot 10^{-10}$	0.0545

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