Multi-Step ODE solvers (PRU01)

Submission deadline: 07.12.2022, 23:59

Edilbert Christhuraj

November 23, 2022

1 Test cases - Initial Value Problems (IVP)

Name	ODE	IC	Domain
IVP0	$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \sin((t + X(t))^2)$	$X(0) = -1$	$t \in [0.0, 4.0]$
IVP1	$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \mathrm{e}^{(t - X(t)\sin(\mathrm{X(t)})}$	$X(0) = 0$	$t \in [0.0, 5.0]$
IVP2	$\frac{\mathrm{d}X^{(1)}(t)}{\mathrm{d}t} = X^{(1)}(t)\left(1 - \alpha X^{(1)}(t)\right) - \frac{X^{(1)}(t)X_2(t)}{1 + \beta X^{(1)}(t)}$ $\frac{\mathrm{d}X^{(2)}(t)}{\mathrm{d}t} = -X^{(2)}(t) + \frac{X^{(1)}(t)X^{(2)}(t)}{1 + \beta X^{(1)}(t)}$ where $\alpha = 0.1,\ \beta = 0.25$	$X^{(1)}(0) = 1$ $X^{(2)}(0) = 0.01$	$t \in [0.0, 65.0]$
IVP3	$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = X(t)^2 - X(t)^3$	$X(0) = 0.005$	$t \in [0, 400]$

2 Numerical schemes

2.1 RK4

The fourth order Runge-Kutta scheme is given by

$$\begin{aligned}
k_1 &= h_i\ F(t_i,\ X_i) \\
k_2 &= h_i\ F\left(t_i + \frac{h_i}{2},\ X_i + \frac{k_1}{2}\right) \\
k_3 &= h_i\ F\left(t_i + \frac{h_i}{2},\ X_i + \frac{k_2}{2}\right) \\
k_4 &= h_i\ F(t_i + h_i,\ X_i + k_3) \\
X_{i+1} &= X_i + (k_1 + 2(k_2 + k_3) + k_4)/6
\end{aligned}$$

where t_i is the i^{th} time step, X_i is the value of X at the i^{th} time step $X(t_i)$ and finally h_i is the value of step size at the i^{th} time step. For RK4, you may take $h_i = h$. It means, for each step we use the same step size.

2.2 RKF45

The fourth order Adaptive Runge-Kutta-Fehlberg is given by

$$\begin{aligned}
k_1 &= h_i\, F(t_i,\, X_i)\\
k_2 &= h_i\, F\left(t_i + \frac{h_i}{4},\, X_i + \frac{k_1}{4}\right)\\
k_3 &= h_i\, F\left(t_i + \frac{3}{8}h_i,\, X_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\right)\\
k_4 &= h_i\, F\left(t_i + \frac{12}{13}h_i,\, X_i + \frac{1932}{2197}k_1 + \frac{-7200}{2197}k_2 + \frac{7296}{2197}k_3\right)\\
k_5 &= h_i * F\left(t_i + h_i, X_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 + \frac{-845}{4104}k_4\right)\\
k_6 &= h_i * F\left(t_i + \frac{1}{2}h_i, X_i + \frac{-8}{27}k_1 + 2k_2 + \frac{-3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)\\
X_{i+1} &= X_i + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6
\end{aligned}$$

where t_i, X_i, h_i are the same as defined in the RK4 subsection. The crucial difference here is that h_i is not constant for all iteration and is chosen adaptively at each iteration according to the following error estimation:

$$h_{\text{new}} = 0.9 * h_i * \left(\frac{\texttt{tolerance}}{\texttt{TruncationError}}\right)^{1/5}$$

where `TruncationError` for RKF45 is given by

$$\texttt{TruncationError} = \left|\frac{1}{360}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6\right|.$$

You can read more about the RKF45 algorithm in `https://en.wikipedia.org/wiki/Runge-Kutta-Fehlberg_method`. Please read it carefully and compare the coefficients in the above scheme with the first table in the Wikipedia page. If you find any inconsistency, please do let me know.

2.3 AB4

The fourth order Adam-Bashforth scheme is given by for $i = 3, 4, \cdots, n-1$

$$X_{i+1} = X_i + h_i\left(\frac{55}{24}F_i - \frac{59}{24}F_{i-1} + \frac{37}{24}F_{i-2} - \frac{9}{24}F_{i-3}\right)$$

where F_i is $F(t_i, X_i)$ and $h_i = h$ and n is the number of steps. You notice that AB4 starts from X_4. In order to calculate X_4, in addition to the initial value X_0 you need the value of X_3 which depends on X_2, X_1 and X_0, and the right hand side function values F_3, F_2, F_1 and F_0. You can use RK4 algorithm to calculate the first three values:X_1, X_2, and X_3. This means, you need to run three RK4 steps to calculate $X_0 \to X_1 \to X_2 \to X_3$. You can use the same step size as RKF45 and the three time step values are: t_0 (given), $t_1 = t_0 + h$, $t_2 = t_1 + h$ and $t_3 = t_2 + h$.

2.4 AB4-AM4 (Predictor-Corrector Model)

The Predictor-Corrector model which uses fourth order Adam-Bashforth and fourth order Adam-Moulton is given by for $i = 3, 4, \cdots, n-1$

Predictor AB4

$$X_{i+1} = X_i + h_i\left(\frac{55}{24}F_i - \frac{59}{24}F_{i-1} + \frac{37}{24}F_{i-2} - \frac{9}{24}F_{i-3}\right)$$

Corrector AM4

$$X_{i+1} = X_i + h_i\left(\frac{9}{24}F_{i+1} + \frac{19}{24}F_i - \frac{5}{24}F_{i-1} + \frac{1}{24}F_{i-2}\right)$$

Similar to AB4 in the previous subsection, you need RK4 to calculate X_3, X_2, X_1 using X_0 and F_0.

3 Tasks

1. Implement the four schemes RK4, RKF45, AB4, AB4-AM4 presented in the previous section as four functions in Python `rk4`, `rkf45`, `ab4`, and `ab4-am4`. Make sure each function takes `ivp` as an input, which is the initial problem (`tStart`, `tEnd`, X_0, and `F`) and returns three arrays, namely time step array `t`$= [t_0, t_1, \cdots, t_n]$, step size array `h`$= [h_0, h_1, h_2, \cdots, h_n]$, and the solution array `x`$= [x_0, x_1, x_2, \cdots, x_n]$.
   For example,

```python
def rk4(ivp):
    ...
    # your code here
    ...
    ...
    return t, h, x

def rkf45(ivp):
    ...
    # your code here
    ...
    ...
    return t, h, x
```

   and so on.

2. Use all the schemes to solve `ivp0` and compare your solutions with the exact solution or inbuilt numerical solvers from Python. Provide a convergence plot in which all 4 algorithms convergence curves are present.

3. Use schemes `rk4` and `rkf45` to solve `ivp1`. What do you observe? First plot the exact solution (which can obtained by inbuilt numerical solvers) and make your observations.

4. Use schemes `rk4`, `rkf45` and `ab4` to solve `ivp2`. What do you observe? How many function evaluations are (approximately) needed in each scheme to obtain the same numerical accuracy (L2 Error, let's say 10^{-5})?

5. Use schemes `ab4` and `ab4-am4` to solve `ivp3`. What happens? Can you solve the `ivp3` using only `am4` (you might need to use iterative solver, fixed point iteration or Newton-like iteration)?

20 **Points**