a.korshuk@innopolis.university
Aliaksei Korshuk
[F21] DE
October 25, 2021

Computational Practicum

Task

Given the initial value problem with the ODE of the first order and some interval:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \\ x \in (x_0, X) \end{cases}$$

Using this application construct a corresponding approximation of the solution of a given initial value problem. Implement the exact solution of an IVP in your application. Provide data visualization capability (charts plotting) in the user interface of your application. Compare approximation errors of these methods plotting the corresponding chart for different grid sizes.

Variant 23

$$\begin{cases} y' = \frac{\sqrt{y-x}}{\sqrt{x}} + 1 \\ y(1) = 10 \\ x \in (1, 15) \end{cases}$$

Source code

GitHub repository is available here.

Exact solution

Initial value problem:

$$y'(x) = \frac{\sqrt{y-x}}{\sqrt{x}} + 1 \tag{1}$$

$$y(1) = 10 \tag{2}$$

Solve $\frac{dy(x)}{dx} = \frac{\sqrt{-x+y(x)}}{\sqrt{x}} + 1$, such that $y(1) = 10$:

Let $y(x) = xv(x)$, which gives $\frac{dy(x)}{dx} = x\frac{dv(x)}{dx} + v(x)$:

$$x\frac{dv(x)}{dx} + v(x) = \frac{\sqrt{-x + xv(x)}}{\sqrt{x}} + 1 \tag{3}$$

Computational Practicum

Simplify:

$$x\frac{dv(x)}{dx} + v(x) = \sqrt{v(x) - 1} + 1 \tag{4}$$

Solve for $\frac{dv(x)}{dx}$:

$$x\frac{dv(x)}{dx} + v(x) = \sqrt{v(x) - 1} + 1 \tag{5}$$

$$\frac{dv(x)}{dx} = \frac{\sqrt{v(x) - 1} - v(x) + 1}{x} \tag{6}$$

Divide both sides by $\sqrt{v(x) - 1} - v(x) + 1$:

$$\frac{\frac{dv(x)}{dx}}{\sqrt{v(x) - 1} - v(x) + 1} = \frac{1}{x} \tag{7}$$

Integrate both sides with respect to x:

$$\int \frac{\frac{dv(x)}{dx}}{\sqrt{v(x) - 1} - v(x) + 1} dx = \int \frac{1}{x} dx \tag{8}$$

Evaluate the integrals, where c_1 is an arbitrary constant:

$$-2\log(-\sqrt{v(x) - 1} + 1) = \log(x) + c_1 \tag{9}$$

Solve for v(x):

$$v(x) = -\frac{2e^{-\frac{c_1}{2}}}{\sqrt{x}} + \frac{e^{-c_1}}{x} + 2 \tag{10}$$

Simplify the arbitrary constants:

$$v(x) = -\frac{2}{c_1\sqrt{x}} + \frac{1}{c_1^2 x} + 2 \tag{11}$$

Substitute back for $y(x) = xv(x)$:

$$y(x) = x(-\frac{2}{c_1\sqrt{x}} + \frac{1}{c_1^2 x} + 2) \tag{12}$$

Solve for c_1 using the initial conditions:
Substitute $y(1) = 10$ into $y(x) = x(-\frac{2}{c_1\sqrt{x}} + \frac{1}{c_1^2 x} + 2)$:

$$-\frac{2}{c_1} + \frac{1}{c_1^2} + 2 = 10 \tag{13}$$

Computational Practicum

Solve the equation:

$$c_1 = -\frac{1}{2} \text{ or } c_1 = \frac{1}{4} \tag{14}$$

Substitute $c_1 = -\frac{1}{2}$ into $y(x) = x(-\frac{2}{c_1\sqrt{x}} + \frac{1}{c_1^2 x} + 2)$:

$$y(x) = 2(2\sqrt{x} + x + 2) \tag{15}$$

Substitute $c_1 = \frac{1}{4}$ into $y(x) = x(-\frac{2}{c_1\sqrt{x}} + \frac{1}{c_1^2 x} + 2)$:

$$y(x) = 2(-4\sqrt{x} + x + 8) \tag{16}$$

After collecting solutions the answer is:

- $y(x) = 2(2\sqrt{x} + x + 2)$
- $y(x) = 2(-4\sqrt{x} + x + 8)$

Results are continuous on their domain.

How does it work?

Input parameters

- Function y: $y = f(x)$
- Function y': $y' = f(x, y)$
- Range of x: $x \in (x_0, X)$
- Initial solution y_0: $y(x_0) = y_0$
- Range of n: $n \in (n_0, N)$
- Coefficient c: one of two possible coefficients after solving initial value problem

Sample results

Input parameters

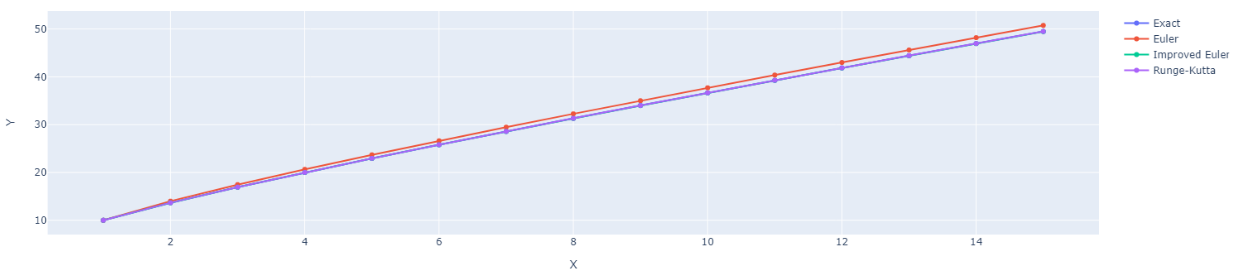
- Function y: $y = x(\frac{-2}{\sqrt{x}c} + \frac{1}{xc^2} + 2)$

Computational Practicum

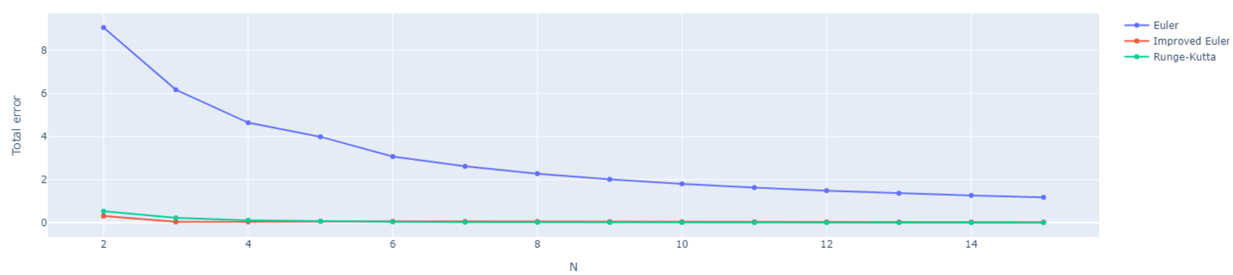
- Function y': $y' = \frac{\sqrt{y-x}}{\sqrt{x}} + 1$
- Range of x: $x \in (1, 15)$
- Initial solution y_0: $y(1) = 10$
- Range of n: $n \in (2, 15)$

Graphs

Exact and numerical solutions



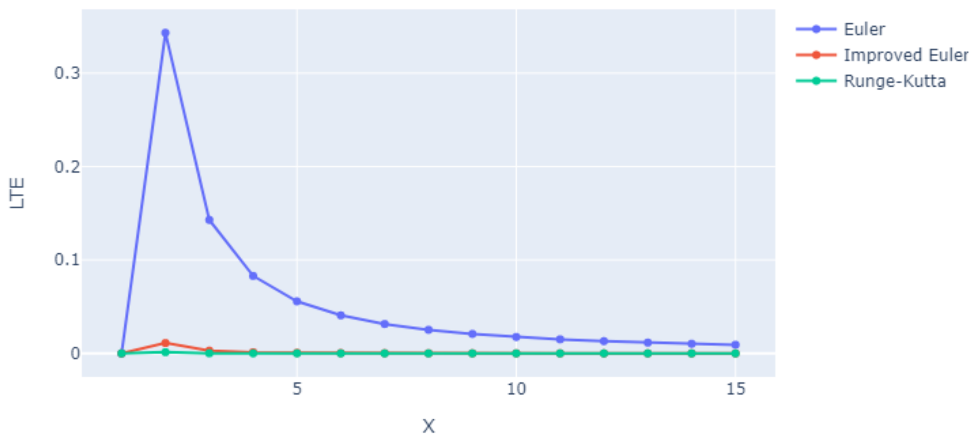
Total approximation error depending on the number of grid cells



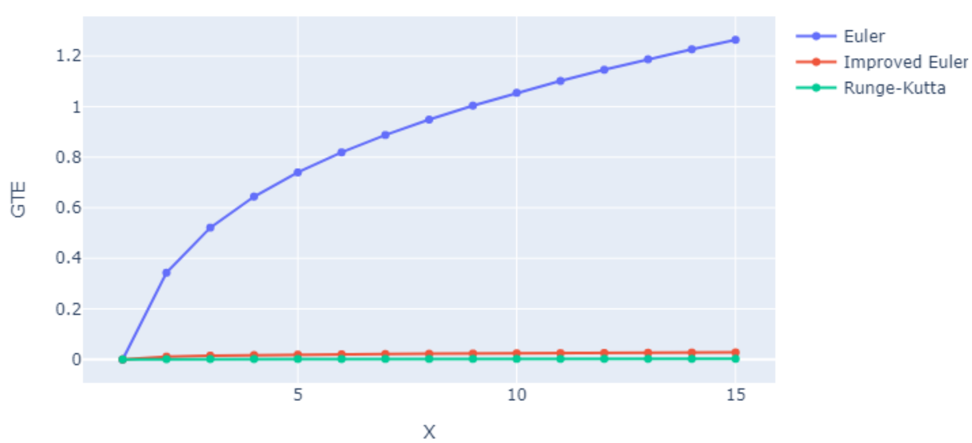
The more points we use, the less total approximation error.

Computational Practicum

Local truncation error



Global truncation error



a.korshuk@innopolis.university
Aliaksei Korshuk

Computational Practicum

Tables

Solutions

	X	Y exact	Y Euler	Y Improved Euler	Y Runge-Kutta
0	1.0	10.0	10.0	10.0	10.0
1	2.0	13.65685424949238	14.0	13.64575131106459	13.658154152609109
2	3.0	16.92820323027551	17.44948974278318	16.913109516716	16.9297524666611
3	4.0	20.0	20.644141555925238	19.982607445908798	20.00168945832998
4	5.0	22.94427190999916	23.684003166177586	22.925232461523127	22.946070704617956
5	6.0	25.79795897113271	26.61708377319725	25.77759917426671	25.799852398622352
6	7.0	28.583005244258363	29.47077755288855	28.561517951870176	28.584984120909258
7	8.0	31.31370849898476	32.262455838458834	31.29122102005047	31.315766222334382
8	9.0	34.0	35.00395145424641	33.976603247930214	34.002131441942026
9	10.0	36.64911064067352	37.703753777859575	36.62487336325267	36.651311628701826
10	11.0	39.2664991614216	40.36819824307491	39.24147561626349	39.26876619514666
11	12.0	41.85640646055102	43.00216158535524	41.83064089084998	41.858736539094025
12	13.0	44.42220510185596	45.60949274770053	44.39573451476892	44.424595610336326
13	14.0	46.96662954709576	48.19329277265756	46.9394855289856	46.96907817766921
14	15.0	49.491933384829665	50.75610371937819	49.464143338702534	49.49443807939822

Local and global truncation errors

	X	LTE Euler	GTE Euler	LTE Improved Euler	GTE Improved Euler	LTE Runge-Kutta	GTE Runge-Kutta
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	2.0	0.3431457505076203	0.3431457505076203	0.011102938427789866	0.011102938427789866	0.0012999031167293396	0.0012999031167293396
2	3.0	0.14286458158996496	0.5212865125076682	0.002960916039452144	0.015093713559512167	0.0001283171101498226	0.0015492363855891256
3	4.0	0.0829037686547629	0.6441415559252377	0.0012168992903198728	0.017392554091202328	2.899280896784262e-05	0.0016894583299809085
4	5.0	0.05572809000084078	0.7397312561784268	0.0006212783840595648	0.01903944847603256	9.631474785720684e-06	0.0017987946187965065
5	6.0	0.04074012986636433	0.8191248020645396	0.0003615240369576611	0.02035979686599987	4.0079016407901236e-06	0.001893427489640942
6	7.0	0.03145030780207492	0.8877723086301863	0.00022965847362144132	0.02148729238818703	1.9342393287047344e-06	0.001978876650895245
7	8.0	0.025225691292057206	0.9487473394740746	0.00015537549787225657	0.022487478934287708	1.0369015051026054e-06	0.00205772334962262
8	9.0	0.020815280171305517	1.0039514542464119	0.00011022928707404844	0.023396752069785975	6.012138626942942e-07	0.002131441942026413
9	10.0	0.01755602599314443	1.054643137186055	8.115469880465298e-05	0.02423727742085191	3.704512820945638e-07	0.002200988028306483
10	11.0	0.015067011285594845	1.101699081653308	6.155475669800126e-05	0.02502354515810623	2.3961558071050604e-07	0.0022670337250616512
11	12.0	0.013115390026108287	1.145755124804218	4.784386986500522e-05	0.02576556970104349	1.612576454590453e-07	0.0023300785430038218
12	13.0	0.01155162788469255	1.1872876458445702	3.795473381984493e-05	0.026470587087040087	1.1216567230576402e-07	0.0023905084803672594
13	14.0	0.010275750985421439	1.2266632255617935	3.063631842081804e-05	0.02714401811016387	8.022654895967207e-08	0.0024486305734470193
14	15.0	0.009218646090950244	1.264170334548524	2.5100313045811617e-05	0.027790046127130097	5.877014075394982e-08	0.0025046945685573974

Computational Practicum

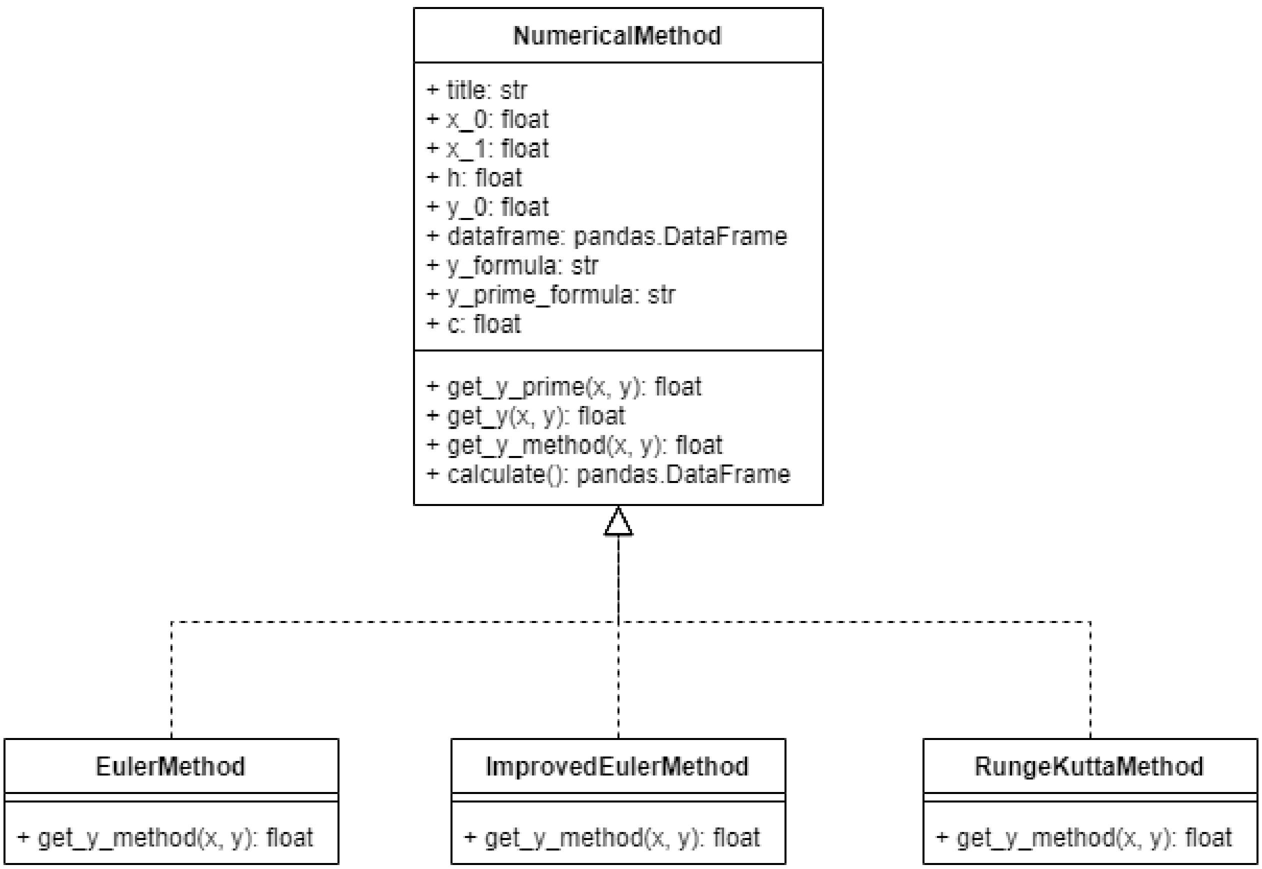
Total errors

	N	Euler	Improved Euler	Runge-Kutta
0	2	9.06350832689629	0.3062324190065624	0.526892894175873
1	3	6.173400561363181	0.03637096898982861	0.22042731474865462
2	4	4.644243242968457	0.038489363908865926	0.11015802275334607
3	5	3.9877038465965526	0.06289426069691473	0.06508617156333685
4	6	3.070015575460488	0.06187719024832461	0.037025365214319095
5	7	2.6157890213594115	0.05860950516954233	0.023620548526231744
6	8	2.2752393901288315	0.05347606695228535	0.01575980411526956
7	9	2.0110171519540785	0.04806884430017533	0.010903155787183039
8	10	1.800398825763402	0.04298178836818778	0.007773125193232033
9	11	1.6287950447145931	0.03840811234109509	0.005683990569309572
10	12	1.486426806133764	0.0343778510990802	0.004247717307784171
11	13	1.3665039233900345	0.030857023515117987	0.0032348792232497203
12	14	1.264170334548524	0.027790046127130097	0.0025046945685573974
13	15	1.175865043673781	0.025117803360203084	0.001967972747493718

Conclusion

All in all, after comparing all methods, we can note, that method Runge-Kutta is the most accurate. Moreover, the more points we use, the less total approximation error.

UML diagram



Toggle callback graph