Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1 по дисциплине «Моделирование»

Тема Решение задачи Коши разными методами

Студент Романов А.В.

Группа ИУ7-63Б

Оценка (баллы) ____________________

Преподаватель Градов В. М.

Москва — 2021 г.

Тема работы

Программная реализация приближенного аналитического метода и численных алгоритмов первого и второго порядков точности при решении задачи Коши для ОДУ.

Цель работы

Получение навыков решения задачи Коши при помощи метода Пикара, метода Эйлера и метода Рунге-Кутты 2-го порядка.

Теоретические сведения

Имеем ОДУ, у которого отсутствует аналитическое решение:

$$\begin{cases} u'(x) = f(x, u) \\ u(\xi) = \eta \end{cases} \tag{1}$$

Для решения данного ОДУ были использованы 3 алгоритма.

Метод Пикара

Имеем:

$$u(x) = \eta + \int_{\xi}^{x} f(t, u(t))\, dt \tag{2}$$

Строим ряд функций:

$$y^{(s)} = \eta + \int_{\xi}^{x} f(t, y^{(s-1)}(t))\, dt, \qquad y^{(0)} = \eta \tag{3}$$

Построим 4 приближения для уравнения (2):

$$y^{(1)}(x) = 0 + \int_{0}^{x} t^2\, dt = \frac{x^3}{3} \tag{4}$$

$$y^{(2)}(x) = 0 + \int_{0}^{x} (t^2 + \left(\frac{t^3}{3}\right)^2)\, dt = \frac{x^3}{3} + \frac{x^7}{63} \tag{5}$$

$$y^{(3)}(x) = 0 + \int_{0}^{x} (t^2 + \left(\frac{t^3}{3} + \frac{t^7}{63}\right)^2)\, dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} \tag{6}$$

$$\begin{aligned} y^{(4)}(x) = 0 + \int_{0}^{x} (t^2 + \left(\frac{t^3}{3} + \frac{t^7}{63} + \frac{2t^{11}}{2079} + \frac{t^{15}}{59535}\right)^2)\, dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \\ \frac{x^{15}}{59535} + \frac{2x^{15}}{93555} + \frac{2x^{19}}{3393495} + \frac{2x^{19}}{2488563} + \frac{2x^{23}}{86266215} + \\ \frac{x^{23}}{99411543} + \frac{2x^{27}}{3341878155} + \frac{x^{31}}{109876902975} \end{aligned} \tag{7}$$

Метод Эйлера

$$y^{(n+1)}(x) = y^{(n)}(x) + h \cdot f(x_n, y^{(n)}) \tag{8}$$

Порядок точности: $O(h)$.

Метод Рунге-Кутта

$$y^{n+1}(x) = y^n(x) + h((1-\alpha)R_1 + \alpha R_2) \tag{9}$$

где $R1 = f(x_n, y^n)$, $R2 = f(x_n + \frac{h}{2\alpha}, y^n + \frac{h}{2\alpha}R_1)$, $\alpha = \frac{1}{2}$ или 1

Порядок точности: $O(h^2)$.

Исходный код алгоритмов

Метод Пикара

Листинг 1: Реализация алгоритма Пикара

```
picard :: Approximation -> Double -> Double
picard 1 x = x ** 3 / 3
picard 2 x = x ** 3 / 3 + x ** 7 / 63
picard 3 x = x ** 3 / 3 + x ** 7 / 63 + 2 * x ** 11 / 2079 + x ** 15 / 59535
picard 4 x = x ** 3 / 3 + x ** 7 / 63 + 2 * x ** 11 / 2079 + x ** 15 / 59535
  + 2 * x ** 15 / 93555 + 2 * x ** 19 / 3393495 + 2 * x ** 19 / 2488563 + 2 * x ** 23 / 86266215
  + x ** 23 / 99411543 + 2 * x ** 27 / 3341878155 + x ** 31 / 109876902975

getPicard :: Approximation -> Double -> Double -> Double -> [Double]
getPicard approx h x0 xm = map (\x -> picard approx x) [x0, x0 + h..xm]
```

Метод Эйлера

Листинг 2: Реализация алгоритма Эйлера

```
euler :: Double -> Double -> Double -> (Double -> Double -> Double) -> Double
euler x y h f = y + h * f x y

getEuler :: Double -> Double -> Double -> Double -> (Double -> Double -> Double) -> [Double]
getEuler h x0 xm y0 f = snd $ foldl
  (\acc x ->
    (euler x (fst acc) h f, snd acc ++ [euler x (fst acc) h f])) (y0, []) [x0, x0 + h..xm]
```

Метод Рунге-Кутта

Листинг 3: Реализация алгоритма Рунге-Кутта

```
rungeKutta :: Double -> Double -> Double -> Alpha -> (Double -> Double -> Double) -> Double
rungeKutta x y h alpha f = y + h * ((1 - alpha) * r1 + alpha * r2)
  where r1 = f x y
        r2 = f (x + h / (2 * alpha)) $ y + h / (2 * alpha) * r1

getRungeKutta :: Double -> Alpha -> Double -> Double -> Double -> (Double -> Double -> Double) -> [Double]
getRungeKutta h alpha x0 xm y0 f = snd $ foldl
  (\acc x ->
    (rungeKutta x (fst acc) h alpha f, snd acc ++ [rungeKutta x (fst acc) h alpha f])) (y0, []) [x0, x0 + h..xm]
```

Результат работы программы

- первый столбец – x;
- второй столбец - метод Пикара (1 приближение);
- третий столбец Пикара (2 приближение);
- четвертый столбец - метод Пикара (3 приближение);
- пятый столбец - метод Пикара (4 приближение);
- шестой столбец - метод Эйлера;
- седьмой столбец - метод Рунге-Кутта.

1.75	1.7864583333333333	2.584316677517361	3.1121015951141233	3.2798154815049942	3.667672171784546	3.6699892007716737
1.76	1.8172586666666666	2.6475836878626944	3.2113308481467047	3.3963177894443177	3.8392358964225592	3.841802195524718
1.77	1.8484110000000002	2.7123285868728817	3.3143016465961734	3.518281198895408	4.0248577869670585	4.027711747308938
1.78	1.8799173333333334	2.778585724861542	3.421179989046141	3.646038227117251	4.226408438287009	4.229596313536429
1.79	1.9117796666666669	2.846390191530295	3.5321397024467722	3.7799450262965046	4.446108810233346	4.449686768502377
1.8	1.9440000000000002	2.9157778285714286	3.647362809568724	3.9203832342147025	4.686616336465906	4.690653175567807
1.81	1.9765803333333336	2.986785242412963	3.7670399131824936	4.067761980360656	4.9511377416044	4.955718528073195
1.82	2.009522666666667	3.059449817106916	3.8913705976881663	4.22252006123224	5.243578757559836	5.248809772186414
1.83	2.0428290000000002	3.1338097273615766	4.020563848949896	4.385128299832273	5.568745629621321	5.574761192685024
1.84	2.0765013333333333	3.2099039517185632	4.154838493119873	4.556092105742207	5.932620589819145	5.939592642224664
1.85	2.1105416666666668	3.2877722858754965	4.294423655268085	4.7359542536660495	6.342745034763192	6.350896845517288
1.86	2.1449520000000004	3.3674553561550638	4.439559238666866	4.925297899987765	6.808762954997942	6.8183891538996395
1.87	2.1797343333333337	3.4489946331212837	4.59049642561312	5.124749858691927	7.343208653364819	7.354705219801903
1.8800000000000001	2.214890666666667	3.532432445343777	4.747498200706238	5.334984159974359	7.962677227324416	7.976587628969686
1.8900000000000001	2.2504230000000005	3.617811993310831	4.910839897536089	5.556725917033218	8.689613888386054	8.706702335881364
1.9000000000000001	2.2863333333333338	3.705177363492065	5.080809769773172	5.790755528899192	9.555140640309865	9.576512725395103
1.9100000000000001	2.3226236666666673	3.7945735425514986	5.2577095876920925	6.03791324975584	10.603696850463566	10.63100695414165
1.9200000000000002	2.3592960000000005	3.8860464317118186	5.441855261199916	6.299104158038654	11.901013970746682	11.936840677661646
1.9300000000000002	2.396352333333334	3.9796428612706456	5.633577490482892	6.575303561707865	13.548598491376254	13.597168042039824
1.9400000000000002	2.4337946666666674	4.0754106052695995	5.8332224454283 2	6.8675628794906505	15.711892958459073	15.780586688500907
1.9500000000000002	2.471625000000001	4.1733983963169665	6.041152475023322	7.177016041611518	18.679993920513525	18.782821027033826
1.9600000000000002	2.509845333333334	4.273655940564766	6.2577468479786615	7.504886457605611	23.00763002817453	23.174378980603386
1.9700000000000002	2.548457666666667	4.37623393284102	6.483402525873881	7.852494603272254	29.910857872934113	30.216283394627823
1.9800000000000002	2.5874640000000007	4.481184071938014	6.718534970169782	8.221266283711996	42.66885538540244	43.35803742216551
1.99	2.6268663333333335	4.588559076057365	6.963578984485753	8.612741634739637	74.19644709239876	76.62332084802506
2.0	2.6666666666666665	4.698412698412698	7.2189895935927675	9.028584930822621	277.36250013756177	327.8931296913306
2.0100000000000002	2.706867000000001	4.810799742990701	7.485242960627818	9.470595274106289	Infinity	NaN
2.02	2.7474693333333335	4.925776080471386	7.762837344092729	9.940718246108116	Infinity	NaN
2.0300000000000002	2.788475666666668	5.043398664308382	8.052294096259173	10.44105861134923	Infinity	NaN
2.04	2.8298880000000004	5.163725546969966	8.354158704662343	10.973894170605726	Infinity	NaN
2.0500000000000003	2.871708333333334	5.2868158963417695	8.669001878429343	11.541690870675355	Infinity	NaN

Рис. 1: Таблица значений с шагом 1e-4, часть значений не выведена

1.999	2.662668666333334	4.6873142447504765	7.192968635224576	8.985853877345333	218.2558920135202	246.95955714655693
1.9991	2.663068286423667	4.688422951692615	7.195565866504818	8.990115266136309	223.01985509346292	253.2104172206030 3
1.9992	2.6634679464960005	4.689531911470519	7.198164177503027	8.99437925364801	227.99404035011784	259.7857535718586
1.9993	2.6638676465523337	4.690641124139432	7.200763568704293	8.998645841737348	233.19256831368395	266.711492152068 9
1.9994	2.6642673865946667	4.691750589754609	7.203364040593928	9.00291503226269	238.63084546539318	274.0163950621223
1.9995	2.664667166625	4.69286030837131	7.205965593657466	9.007186827083853	244.32571330617102	281.732459070916
1.9996	2.6650669866453334	4.693970280044812	7.2085682283806 6	9.011461228062117	250.29561856444394	289.8953832368011
1.9997	2.6654668466576665	4.6950805048304005	7.211171945249481	9.0157382370602	256.5608081117087	298.5451200188733
1.9998	2.665866746664	4.6961909827833725	7.2137767447501275	9.020017855942307	263.143552857606	307.7265277916339
1.9999	2.6662666866663334	4.697301713959032	7.216382627369011	9.024300086574076	270.06840575865937	317.4901472052287 7
2.0	2.6666666666666665	4.698412698412698	7.2189895935927675	9.028584930822621	277.36250013756177	327.8931296913306
2.0001	2.6670666866670008	4.699523936199702	7.221597643908261	9.032872390556527	285.05589582581865	339.0003540495301
2.0002	2.667466746669333	4.700635427375372	7.224206778802546	9.037162467645791	293.18198228032867	350.8857770802122
2.0003	2.667866846675667	4.70174717199507	7.226816998762948	9.041455163961961	301.77794987372	363.6340775225411
2.0004	2.6682669866879998	4.702859170114145	7.229428304276961	9.045750481377954	310.88534313673455	377.34267033182164
2.0005	2.668667166708334	4.703971421787978	7.2320406958323495	9.050048421768246	320.55071299448406	392.1241923329345
2.0006	2.6690673867386665	4.70508392707194	7.234654173917051	9.054348987008693	330.8263891946473	408.1095930351305
2.0007	2.669467646781001	4.706196686021434	7.237268739019272	9.058652178976706	341.7713994534531	425.4520095750348
2.0008	2.6698679468373334	4.7073096986918515	7.239884391627394	9.06295799955108	353.4525687219543	444.3316678389043
2.0009	2.670268286909667	4.708422965138617	7.242501132230063	9.067266450612168	365.9458409156501	464.9621410371953 5
2.001	2.6706686669999993	4.709536485417143	7.245118961316107	9.07157753404171	379.3378771640963	487.5984249619726
2.0011	2.6710690871103338	4.710650259582875	7.247737879374617	9.075891251723004	393.7280001093536	512.5474754557406

Рис. 2: Таблица значений с шагом 1e-4 на интервале [1.999, 2.0011]

Ответы на вопросы

Укажите интервалы значений аргумента, в которых можно считать решением заданного уравнения каждое из первых 4-х приближений Пикара. Точность результата оценивать до второй цифры после запятой. Объяснить свой ответ.

Мы можем считать первое приближение решением уравнения до того момента пока значения первого и второго приближения совпадают до второй цифры после запятой. Если приближения расходятся, нужно рассматривать второе и третье приближение по такому же принципу – если значения сходятся, то решением на таком интервале является второе приближение. И так далее.

- На интервале [0; 0.75] решением можно считать первое приближение.
- На интервале (0.75; 1.22] решением можно считать второе приближение.
- На интервале (1.22; 1.48] решением можно считать третье приближение.
- Для нахождения интервала, где решением можно считать четвертое приближение, нужно рассчитать пятое приближение.

Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.

Для того чтобы доказать правильность полученного результата, нужно рассмотреть значения в окрестности точки, с шагом приближенным к нулю. Если при таком шаге значения примерно совпадают, то значение посчитано правильно.

Каково значение функции при x = 2, т.е. привести значение u(2).

Около 327.