

*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”*

Лабораторна робота №12

з курсу «Чисельні методи»

***Тема: «Лінійні багатокрокові різницеві явні і неявні методи рішення рівнянь задачі
Коші для звичайних диференціальних рівнянь»***

Виконав:
студент II курсу
групи ДА-72
Кондратюк Тарас
Варіант 15

Київ - 2019

ЗАВДАННЯ

№	$y' = f(t, y)$	t_0	t_k	y_0
15	$10y/(1-t^2)$	0	10	1

ВИКОНАННЯ РОБОТИ

$f[t_, y_] := y * \text{Exp}[-2 * t];$

- Програмувати на мові пакету *Mathematica* рішення заданого диференційного рівняння лінійним явним багатокроковим різницеvim методом третього порядку і експериментально визначаємо максимально можливий крок h_{max} сталого рішення.

```
LTSEDMofT0[f_, t0_, u0_, tm_, m_] := Block[{t = t0, u = u0, h = (tm - t0)/m,
X = Array[x, m + 1, 0]; x[0] = t0;
Y = Array[y, m + 1, 0]; y[0] = u0;
k1[h_] := f[t, u];
k2[h_] := f[t + 0.5 * h, u + 0.5 * h * k1[h]];
k3[h_] := f[t + 0.5 * h, u + 0.5 * h * k2[h]];
k4[h_] := f[t + h, u + h * k3[h]];
du = h * (k1[h] + 2 * k2[h] + 2 * k3[h] + k4[h]) / 6; t = t + h; u = u + du; x[1] = t; y[1] = u;
If[m > 1,
Do[y[i + 1] = 5 * y[i - 1] - 4 * y[i] + 2 * h * (f[x[i - 1], y[i - 1]] + 2 * f[x[i], y[i]]);
x[i + 1] = x[i] + h, {i, 1, m - 1}];];
TableForm[{X, Y}, TableHeadings -> {{t, u}, Table[j, {j, 0, m}]},
TableSpacing -> {4, 2}, TableAlignments -> Center]
];
```

LTSEDMofT0[f, 0, 1, 10, 10]

bleForm=

	0	1	2	3	4	5	6	7	8	9	10
t	0	1	2	3	4	5	6	7	8	9	10
u	1	1.54021	1.67294	1.54874	2.24636	-1.23101	16.1571	-70.7832	363.918	-1809.59	9057.94

Як видно, результат розбігається. Збільшимо кількість вузлових точок, тобто зменшимо крок.

LTSEDMofT0[f, 0, 1, 10, 100 000] Як видно, при кроці 0.0001 метод все ще розбігається.

y[100 000] - y[99 999]
 $1.376043467597148 \times 10^{69880}$

- Порівнюємо отриманий максимально можливий крок h_{max} з значеннями, обчисленим за допомогою формули (4.13).

$$h = \frac{c}{\lambda_{max}} = \frac{2.7853}{\max\left(\frac{\partial f}{\partial y}\right)} = \frac{2.7853}{\max(e^{-2t})} = 2.7853$$

- Програмуємо рішення заданого диференційного рівняння неявним методом Гіра третього порядку і виконуємо рішення при значеннях кроку, вдвічі більшого ніж значення, отримане з пункту 1.

```
MethodGira3[f_, t0_, u0_, tm_, m_] := Block[{t = t0, u = u0, h = (tm - t0)/m},
  X = Array[x, m + 1, 0]; x[0] = t0;
  Y = Array[y, m + 1, 0]; y[0] = u0;
  Do[x[i] = t0 + i * h, {i, 1, m}];
  k1[h_] := f[t, u];
  k2[h_] := f[t + 0.5 * h, u + 0.5 * h * k1[h]];
  k3[h_] := f[t + 0.5 * h, u + 0.5 * h * k2[h]];
  k4[h_] := f[t + h, u + h * k3[h]];
  Do[du = h * (k1[h] + 2 * k2[h] + 2 * k3[h] + k4[h]) / 6; t = t + h; u = u + du; y[i] = u, {i, 2}];
  If[m > 2, Do[root = Solve[temp == (1/11) (18 y[i] - 9 y[i - 1] + 2 y[i - 2] + 6 * h * f[x[i + 1], temp])];
    y[i + 1] = temp /. root[[1]], {i, 2, m - 1}];];
  TableForm[{X, Y}, TableHeadings -> {{t, u}, Table[j, {j, 0, m}]}, TableSpacing -> {4, 2},
    TableAlignments -> Center]
];
```

MethodGira3[f, 0, 1, 10, 100]

TableForm=

	0	1	2	3	4	5	6	7	8	9	10
t	0	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{2}{10}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{7}{10}$	$\frac{4}{5}$	$\frac{9}{10}$	1
u	1	1.09487	1.1792	1.25314	1.31713	1.37194	1.41849	1.45776	1.4907	1.51821	1.5411
	11	12	13	14	15	16	17	18	19	20	
	$\frac{11}{10}$	$\frac{6}{5}$	$\frac{13}{10}$	$\frac{7}{5}$	$\frac{3}{2}$	$\frac{8}{5}$	$\frac{17}{10}$	$\frac{9}{5}$	$\frac{19}{10}$	2	
	1.5601	1.57582	1.58881	1.59952	1.60834	1.6156	1.62157	1.62647	1.6305	1.6338	

90	91	92	93	94	95	96	97	98	99	100
9	$\frac{91}{10}$	$\frac{46}{5}$	$\frac{92}{10}$	$\frac{47}{5}$	$\frac{19}{2}$	$\frac{48}{5}$	$\frac{97}{10}$	$\frac{49}{5}$	$\frac{99}{10}$	10
1.64879	1.64879	1.64879	1.64879	1.64879	1.64879	1.64879	1.64879	1.64879	1.64879	1.64879

4. Програмуємо на мові пакету *Mathematica* рішення заданої системи диференціальних рівнянь неявним методом Гіра четвертого порядку і виконуємо рішення при кроці h_{max} , використаному в пункті 3.

```
MethodGira4[f_, t0_, u0_, tm_, m_] := Block[{t = t0, u = u0, h =  $\frac{tm - t0}{m}$ },
  X = Array[x, m + 1, 0]; x[0] = t0;
  Y = Array[y, m + 1, 0]; y[0] = u0;
  Do[x[i] = t0 + i * h, {i, 1, m}];
  k1[h_] := f[t, u];
  k2[h_] := f[t + 0.5 * h, u + 0.5 * h * k1[h]];
  k3[h_] := f[t + 0.5 * h, u + 0.5 * h * k2[h]];
  k4[h_] := f[t + h, u + h * k3[h]];
  Do[du = h * (k1[h] + 2 * k2[h] + 2 * k3[h] + k4[h]) / 6; t = t + h; u = u + du; y[i] = u, {i, 3}];
  If[m > 3,
    Do[root = Solve[temp ==  $\frac{1}{25}$  (48 y[i] - 36 y[i - 1] + 16 y[i - 2] - 3 y[i - 3] + 12 * h * f[x[i + 1], temp))];
      y[i + 1] = temp /. root[[1]], {i, 3, m - 1}];];
  TableForm[{X, Y}, TableHeadings -> {{t, u}}, Table[j, {j, 0, m}], TableSpacing -> {4, 1},
    TableAlignments -> Center]
];
```

MethodGira4[f, 0, 1, 10, 100]

TableForm=

	0	1	2	3	4	5	6	7	8	9	10
t	0	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{2}{10}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{7}{10}$	$\frac{4}{5}$	$\frac{9}{10}$	1
u	1	1.09487	1.1792	1.25307	1.31695	1.37166	1.41814	1.45737	1.4903	1.51781	1.54072
	90	91	92	93	94	95	96	97	98	99	100
	9	$\frac{91}{10}$	$\frac{46}{5}$	$\frac{92}{10}$	$\frac{47}{5}$	$\frac{19}{2}$	$\frac{48}{5}$	$\frac{97}{10}$	$\frac{49}{5}$	$\frac{99}{10}$	10
	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486	1.6486

5. Користуючись стандартними операторами пакету *Mathematica*, знаходимо рішення заданого диференціального рівняння неявним методом Брайтона.

```
Y = NDSolve[{y'[t] == y[t] * Exp[-2 * t], y[0] == 1}, y, {t, 0, 10}, Method -> "BDF"]
{{y -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
y[t_] = y[t] /. Y[[1]];
```

	0.	0.5	1.	1.5	2.	2.5	3.	3.5	4.	4.5	5.
t	0.	0.5	1.	1.5	2.	2.5	3.	3.5	4.	4.5	5.
u	1.	1.37171	1.54085	1.60819	1.63369	1.64318	1.64668	1.64797	1.64844	1.64862	1.64868
	5.	5.5	6.	6.5	7.	7.5	8.	8.5	9.	9.5	10.
	5.	5.5	6.	6.5	7.	7.5	8.	8.5	9.	9.5	10.
	1.64868	1.64871	1.64872	1.64872	1.64872	1.64872	1.64872	1.64872	1.64872	1.64872	1.64872

6. Звіт :

В даній лабораторній роботі було використано лінійні багатокрокові різницеві явні і неявні методи рішення рівнянь задачі Коші для звичайних диференціальних рівнянь, а саме: лінійний явний багатокроковий (двохкроковий) різницевий метод третього порядку, неявний метод Гіра 3-го та 4-го порядків, а також за допомогою стандартного оператора пакету Mathematica, задане диференціальне рівняння було розв'язане методом Брайтона

t	Неявний Гіра 3	Неявний Гіра 4	Mathematica(BDF)
0	1	1	1
0.5	1.3719434668683694	1.3716633445328907	1.3717128308664899
1	1.5411007796177978	1.5407198062928786	1.540847116130692
1.5	1.6083440135586626	1.6080561635201716	1.6081855776114677
2	1.6337970142552098	1.6335676822387424	1.6336915978149038
2.5	1.6432593697233333	1.6430552784106427	1.6431761604118165
3	1.6467538309115812	1.6465595505749941	1.64667914846147
3.5	1.648041197037881	1.6478505982780105	1.6479696856907038
4	1.6485150398866981	1.648325805352592	1.6484446876194843
4.5	1.6486893904117148	1.648500659084247	1.6486194505853662
5	1.6487535349173892	1.6485649888902578	1.648683735076442
5.5	1.6487771329756538	1.6485886551412035	1.6487073820579055
6	1.6487858142991547	1.6485973615547052	1.6487160693221852
6.5	1.6487890079908267	1.6486005644769244	1.6487192656603666
7	1.6487901828858549	1.648601742767742	1.6487204312610677
7.5	1.6487906151057872	1.6486021762369236	1.6487208586080022
8	1.6487907741106433	1.6486023357013528	1.648721015573748
8.5	1.6487908326052645	1.6486023943650412	1.6487210641529813
9	1.6487908541242353	1.6486024159462076	1.648721077989691
9.5	1.6487908620406209	1.648602423885475	1.6487210848960494
10	1.6487908649528977	1.6486024268061683	1.6487210879636351

З таблиць результатів видно, що значення отримані неявним методом Гіра 3-м та 4-м порядком відрізняються один від одного лише у 4-му знаці після коми. Причому, варто зауважити, що значення отримане неявним методом Гіра 3-го порядку апроксимації виявилось точнішим за значення отримане методом 4-го порядку

апроксимації. Це можна пояснити лише тим, що ці методи зі зростанням порядку стають менш стійкими, на відміну від методів Рунге-Кутта. У зв'язку з цим, очевидно, потрібно зменшити крок, для отримання більш точного результату за допомогою методу 4-го порядку апроксимації, ніж за допомогою 3-го порядку апроксимації.