Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика
Катедра “Приложна математика и моделиране”

“Компютърни числени методи”
ЛЕКЦИЯ 9

Проф. д-р Снежана Гочева-Илиева, snow@uni-plovdiv.bg

Он-лайн обучение: www.fmi-plovdiv.org/evlm
www.fmi-plovdiv.org/evlm/DBbg - числени методи

Литература:

1. Бояджиев Д., Гочева С., Макрелов И., Попова Л. – Ръководство по числени методи – част 1, Издания: 2003, 2006, 2010.
2. Семерджиев Х., Боянов Б., Числени методи, ПУ.
3. Гочева-Илиева С., Въведение в система Mathematica, ЕксПрес, Габрово, 2009.

Съдържание

Числени методи за обикновени диференциални уравнения (ОДУ)

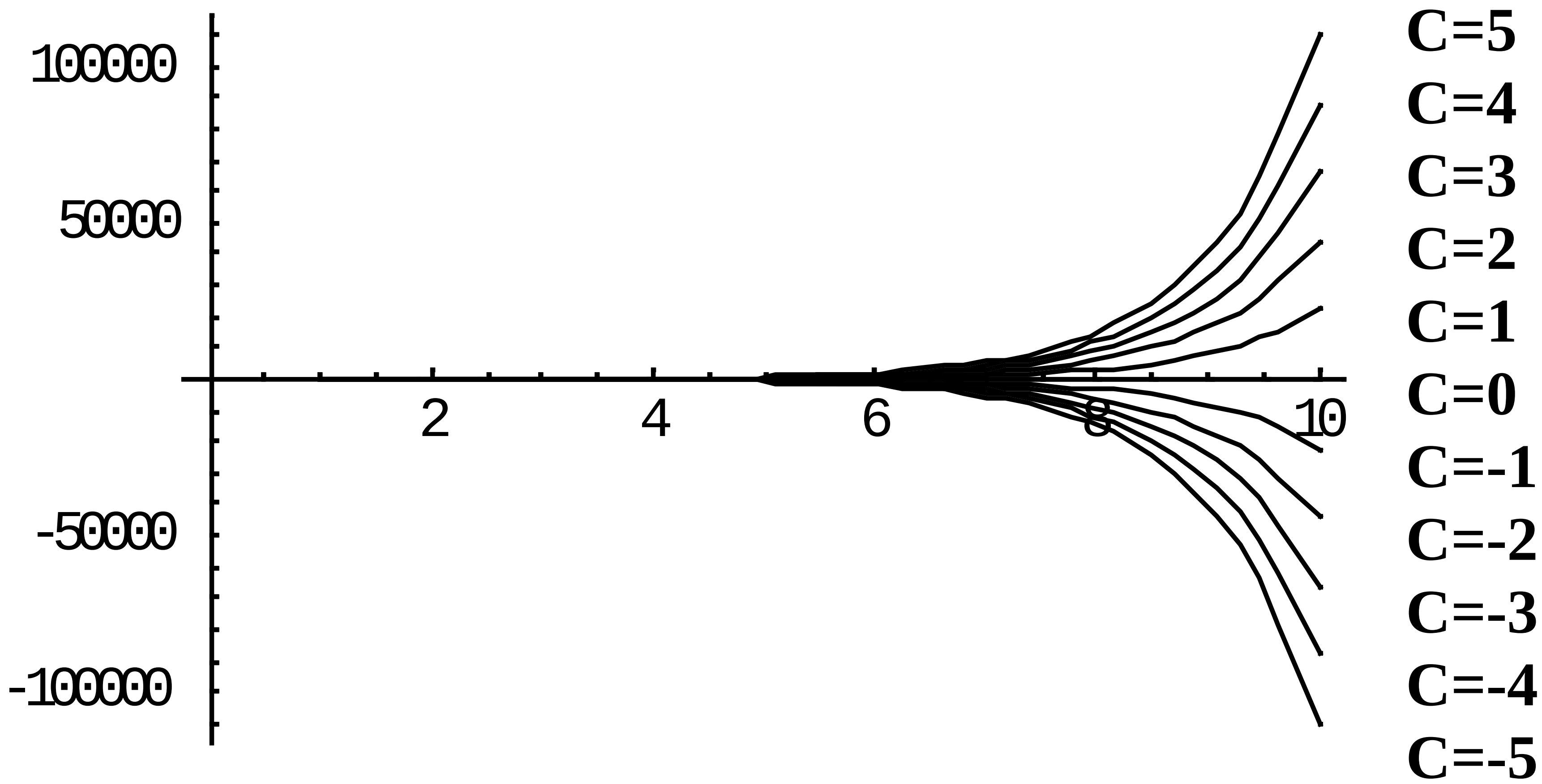
1. Преговор по ОДУ

Определение 1. Диференциално уравнение е уравнение, в което неизвестното е функция, заедно с производните й до някакъв ред. *Определение 2.* Ако неизвестната функция зависи само от една променлива, уравнението се нарича обикновено диференциално уравнение (ОДУ). Ако независимите променливи са две или повече, говорим за частно диференциално уравнение (ЧДУ).

Пример 1. ОДУ от първи ред е например уравнението:

$$y'(x) = y(x), \tag{1}$$

където неизвестната функция $y(x)$ се търси в някакъв интервал $[x_0, x_0 + a]$. Очевидно едно решение на това уравнение е функцията $y(x) = e^x$. Очевидно е също, че и за всяка константа C, $y(x) = Ce^x$ също е решение, т.е. има безброй решения на уравнение (1).



Фиг. 1. Графика на част от общото решение на (1): $y(x) = Ce^{x}$ за C=-5, -4, ..., 5 в интервала $[1,10]$.

Определение 3. Множеството от всички решения на едно ОДУ се нарича ***общо решение***. От теорията на ОДУ е известно, че общото решение на ОДУ от 1-ви ред зависи от една произволна константа, общото решение на ОДУ от 2-ри ред зависи от две произволни константи, от трети ред- от три константи и т.н.

Определение 4. Всяко отделно решение, което се получава от общото при конкретна стойност на константите, се нарича ***частно решение*** на даденото ОДУ.

Пример 2. ОДУ от 2-ри ред е уравнението: $y''(x)-\dfrac{x.y'(x)-y(x)}{x^2+\sin(y(x))}=0$.

Пример 3. Частно диференциално уравнение от втори ред:

$\dfrac{\partial U}{\partial t}=\dfrac{\partial^2 U}{\partial x^2}+\dfrac{\partial^2 U}{\partial y^2}$ с неизвестна ф-я $U=U(x,y,t)$ (параболично ЧДУ).

2. Постановка на задачата на Коши (начална задача за ОДУ)

Търси се решението $y = y(x)$ на задачата:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \quad , \quad x \in [x_0, b]. \end{cases} \qquad (2)$$

Вторият ред на задачата се нарича начално условие, тъй като от общото решение се търси онова частно решение, което в точката x_0 има дадена стойност $y(x_0) = y_0$.

<u>Теорема на Коши.</u> Ако дясната част $f(x, y)$ на уравнението (2) и частната ѝ производна $\dfrac{\partial f(x, y)}{\partial y}$ са дефинирани и непрекъснати в област $G(x, y) \in \mathbb{R}^2$, то за всяка вътрешна точка $(x_0, y_0) \in G$ съществува единствено решение $y(x)$ на задача (2).

По-нататък ще считаме, че решението на (2) съществува.

3. Класификация на задачите за ОДУ

Могат да се формулират следните основни задачи:

А) задача на Коши за ОДУ от 1-ви ред

Б) задача на Коши за ОДУ от *p*-ти ред

В) задача на Коши за система ОДУ от първи ред

Г) гранична задача за ОДУ от 2-ри ред

Първата задача вече дефинирахме с (2). Тя може да се зададе и в т.н. неявен вид:

$$\left| \begin{array}{l} f(x, y, y') = 0 \\ \varphi(x_0, y(x_0)) = 0, \quad x \in [x_0, b]. \end{array} \right. \tag{3}$$

Задача Б) може също да е в явен или неявен вид.

Напр.: Търси се решението на ОДУ (явен вид)

$$\left| \begin{array}{l} y^{(p)} = f(x, y, y', ..., y^{(p-1)}) \\ y(x_0) = y_0,\ y'(x_0) = y'_0,\ ...,\ y^{(p-1)}(x_0) = y_0^{(p-1)}, \quad x \in [x_0, b]. \end{array} \right. \tag{4}$$

<u>В) - задача на Коши</u> за система ОДУ от първи ред е:

Търсят се неизвестните функции $y_1(x), y_2(x), ..., y_p(x)$, които удовлетворяват системата уравнения:

$$\left| \begin{array}{l} y'_1 = f_1(x, y_1, y_2, ..., y_p) \\ y'_2 = f_2(x, y_1, y_2, ..., y_p) \\ ... \\ y'_p = f_p(x, y_1, y_2, ..., y_p) \end{array} \right., \quad x \in [x_0, b]. \tag{5a}$$

и граничните условия:

$$\left|\begin{array}{l} y_1(x_0)=\alpha_1 \\ y_2(x_0)=\alpha_2 \\ \ldots \\ y_p(x_0)=\alpha_p \end{array}\right. . \qquad (5б)$$

По-нататък задачата (5а,б) ще представяме във векорен вид, аналогичен на (2):

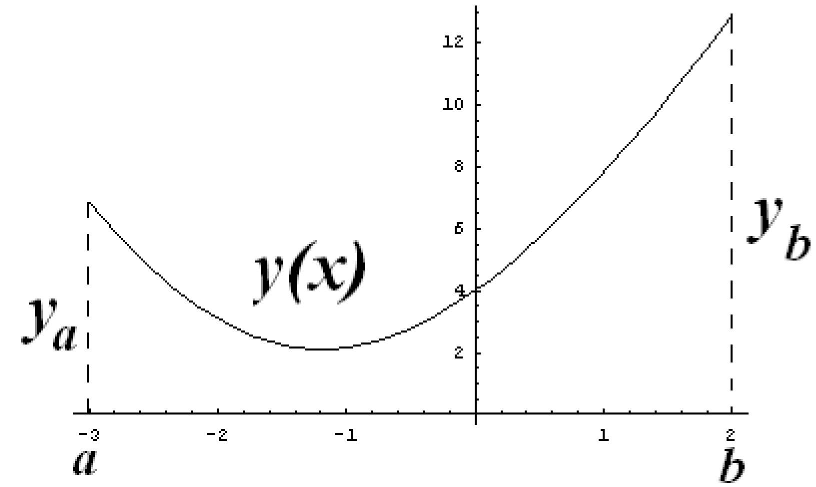
$$\begin{array}{l} Y'=F(x,Y) \\ Y(x_0)=\mathrm{A} \quad , \quad x\in[x_0,b] , \end{array} \qquad (6)$$

където сме означили векторите с

$$Y=\begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_p \end{pmatrix}, \quad F=\begin{pmatrix} f_1(x,y_1,y_2,\ldots,y_p) \\ f_2(x,y_1,y_2,\ldots,y_p) \\ \ldots \\ f_p(x,y_1,y_2,\ldots,y_p) \end{pmatrix}, \quad A=\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_p \end{pmatrix}.$$

Идея за задача Г) - гранична задача за ОДУ от 2-ри ред:

$$\left| \begin{array}{l} y'' = f(x, y, y'), \qquad x \in (a, b) \\ y(a) = y_a, \quad y(b) = y_b \end{array} \right. .$$



4. Свеждане на задача за уравнение от по-висок ред към система ОДУ.

За простота ще разгледаме случая за уравнение от 2-ри ред:

$$\left| \begin{array}{l} y'' = f(x, y, y'), \qquad x \in [x_0, b] \\ y(x_0) = \alpha_1 \\ y'(x_0) = \alpha_2 \end{array} \right. .$$

Полагаме: $\boxed{y'(x) = z(x)}$. Тогава $y''(x) = z'(x)$ и получаваме веднага начална задача за системата ОДУ:

$$\left| \begin{array}{l} y' = z, \\ z' = f(x, y, z), \qquad x \in [x_0, b] \\ y(x_0) = \alpha_1 \\ z(x_0) = \alpha_2 \end{array} \right. .$$

или във векторен вид:

$$\left\|\begin{array}{l} \begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} z \\ f(x, y, z) \end{pmatrix}, \qquad [x_0, b] \\ \begin{pmatrix} y(x_0) \\ z(x_0) \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \end{array}\right. .$$

5. Едностъпкови методи. Метод на Ойлер. Оценка на локалната грешка. Пример.

Числените методи решават началната задача на Коши като търсят **таблица на неизвестната функция в избрано множество (мрежа) от точки в зададения интервал**. При необходимост по-нататък се интерполира или се прави друг тип приближение за намиране на приближена формула на решението.

Метод на Ойлер (1768 г.)- явен едностъпков метод

Решава се задачата на Коши:

$$\left| \begin{array}{l} y' = f(x, y) \\ y(x_0) = y_0 \end{array} \right. , \quad x \in [x_0, b]. \qquad (2)$$

В интервала $[x_0, b]$ се и избират *n* точки: $x_0, x_1, ..., x_{n-1}, x_n = b$. Най-често това става равномерно, като се изчислява стъпка $h = (b - x_0)/n$ и точките се намират по формулата: $x_i = x_0 + ih, \quad i = 0, 1, 2, ..., n$. Тогава $x_1 - x_0 = h, \quad x_2 - x_1 = h, ...$

Извод на формулата за метода на Ойлер:

Развиваме функцията $y(x)$ в $x_1 = x_0 + h$ по формулата на Тейлър около точката x_0. Имаме:

$$y(x_1) = y(x_0 + h) = y(x_0) + h\, y'(x_0) + \frac{h^2}{2!} y''(\xi), \quad \xi \in (x_0, x_1). \qquad (8)$$

Тъй като търсим функция, която удовлетворява уравнението и началното условие в (2), то $y'(x_0) = f(x_0, y_0)$ и $y(x_0) = y_0$. Получаваме:

$$y(x_1) = y_0 + hf(x_0, y_0) + \frac{h^2}{2!} y''(\xi), \quad \xi \in (x_0, x_1).$$

Ако тук последното събираемо разглеждаме като локална грешка за точката x_0, намираме приближена формула за $y(x_1)$:

$$\boxed{y(x_1) = y_0 + hf(x_0, y_0)}. \tag{9}$$

Следователно като знаем решението в (x_0, y_0) от (9) намираме решението в съседната точка (x_1, y_1), като сме означили $y_1 = y(x_1)$. Грешката на този етап при ограничена втора производна е:

$R(x) = \frac{h^2}{2!} y''(\xi),$ или оценена отгоре е: $\left|R(x)\right| \le \frac{h^2}{2!} \max_{[x_0, x_1]} \left|y''(\xi)\right| = O(h^2)$.(10)

Обща формула за метода на Ойлер (стъпка след стъпка):

$$\boxed{y(x_{i+1}) = y_i + hf(x_i, y_i)}, \quad i = 0, 1, \ldots, n-1. \tag{11}$$

Пример 4.

Задача. Да се реши по метода на Ойлер началната задача : $y' = \sqrt{x+y} + y\,\mathrm{Cos}[x\,y],\quad y(1) = 1$ **в интервала [1, 2] и стъпка h = 0.05 .**

<u>Решение</u> : Дефинираме функцията по правилата на система *Mathematica* и програмираме :

```
f[x_, y_] := Sqrt[x + y] + y Cos[x y]
n = 20; a = 1.; b = 2; h = (b - a)/n;
```

```
x = a; y = 1.;
res = {{"i", "x", "y", "f(x,y)"}, {0, x, y, f[x, y] }};
For[i = 0, i < n, i++, fn = f[x, y];
    x = x + h; y = y + h * fn;              (* Формула на Ойлер *)
    res = Append[res, {i + 1, x, y, fn}]]
TableForm[res]
```

i	x	**y**	f(x,y)
0	1.	**1.**	1.95452
1	1.05	**1.09773**	1.95452
2	1.1	**1.19329**	1.9113
3	1.15	**1.28424**	1.81903
4	1.2	**1.36827**	1.68064
5	1.25	**1.44354**	1.50534
6	1.3	**1.50889**	1.30701
7	1.35	**1.56395**	1.10125
8	1.4	**1.60906**	0.90222
9	1.45	**1.64509**	0.720528
10	1.5	**1.67322**	0.562579
11	1.55	**1.69477**	0.431085

12	1.6	**1.71108**	0.326119
13	1.65	**1.72339**	0.246214
14	1.7	**1.73285**	0.189244
15	1.75	**1.74051**	0.153034
16	1.8	**1.74729**	0.13575
17	1.85	**1.7541**	0.136144
18	1.9	**1.76179**	0.153732
19	1.95	**1.77123**	0.188971
20	2.	**1.78341**	0.243499

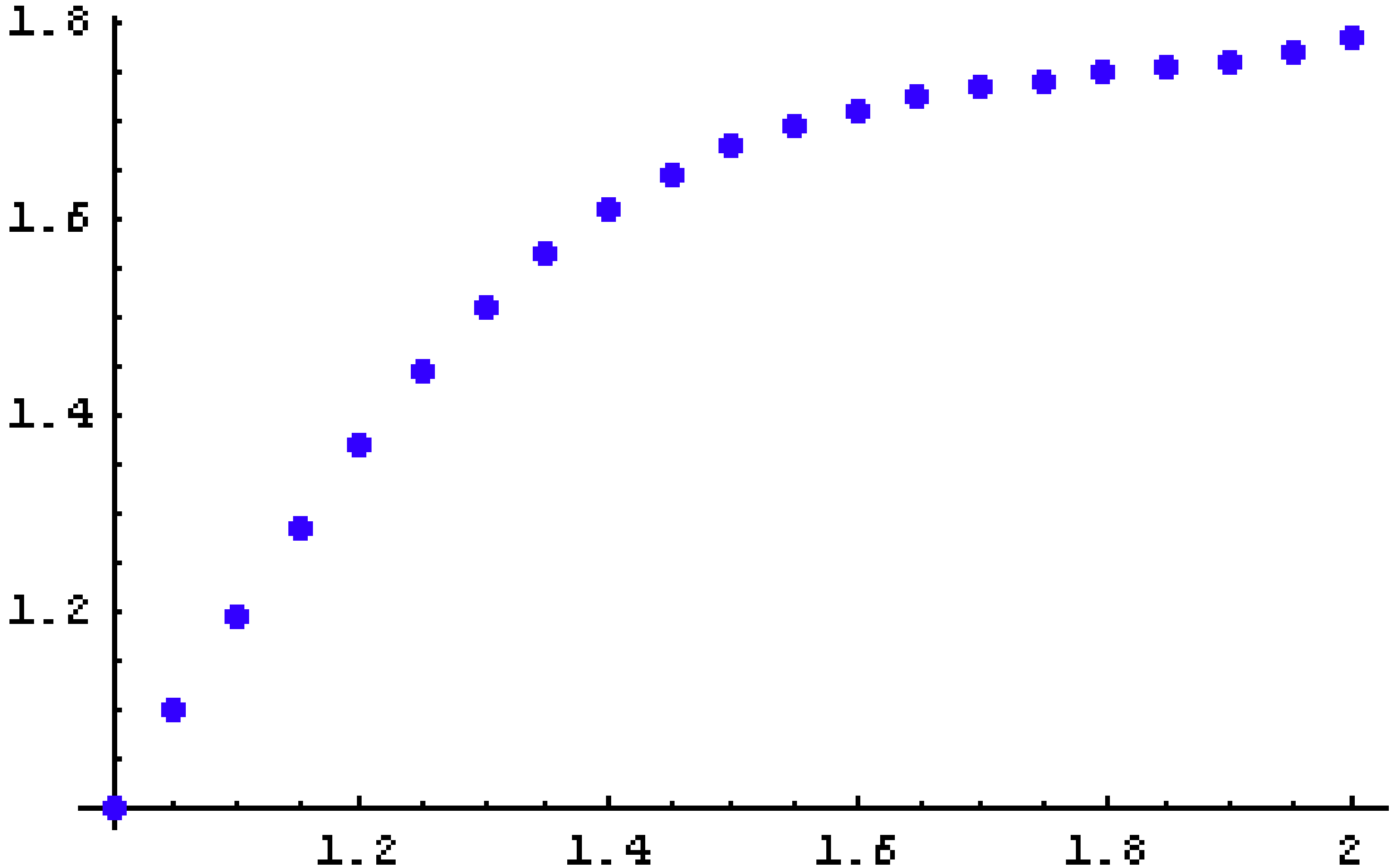
Заключение: **Решението по метода на Ойлер е във вид на таблица на неизвестната функция y(x) - съответно втора колона от горната таблица. Тъй като теоретичната локална грешка на метода е O(h^2) и тук h=0.05, то тези решения трябва да се вземат закръглени само с три знака след десетичната точка. За уточняване може да се използва принципа на Рунге.**

Графика на решението

```
res2 = Drop[res, 1]; res2 = res2[[All, {2, 3} ]]; TableForm[res2]
  (* Извличат се само вторите и третите компоненти от всички елементи
   на списъка, т.е. {x,y}*)
 gr1 = ListPlot[res2, PlotStyle → {Hue[0.7], PointSize[0.02]}]
  (* Чертае се точковата графика на решението y (x) *)
```

1.	1.
1.05	1.09773
1.1	1.19329
1.15	1.28424
1.2	1.36827
1.25	1.44354
1.3	1.50889
1.35	1.56395

1.4	1.60906
1.45	1.64509
1.5	1.67322
1.55	1.69477
1.6	1.71108
1.65	1.72339
1.7	1.73285
1.75	1.74051
1.8	1.74729
1.85	1.7541
1.9	1.76179
1.95	1.77123
2.	1.78341


1.8
1.6
1.4
1.2
1.2
1.4
1.6
1.8
2

Пример 5.

Дадено : Начална задача за система от обикновени диференциални уравнения от вида :

$$y' = f(x, y, z),\ z' = g(x, y, z),\ x \in [a, b],\ y(a) = y_0,\ z(a) = z_0.$$

Цел : Да се намери приближено решение

$\{y(x),\ z(x)\}$ на задачата по метода на Ойлер при зададена стъпка h.

Теория : Това е едностъпков метод,

при който от решението $\{y_i,\ z_i\}$ в една точка x_i се изчислява решението $\{y_{i+1},\ z_{i+1}\}$

в съседната точка $x_{i+1} = x_i + h$ по формулите

$y_{i+1} = y_i + h.f(x_i, y_i, z_i),\ z_{i+1} = z_i + h.g(x_i, y_i, z_i),\ i = 0, 1, \ldots$ и $\{y_0,\ z_0\}$ са дадените начални стойности.

Задача. Да се реши системата : $y' = x + y + z^2, \; z' = \frac{y+z}{1+x^2}, \; y(1) = 1, \; z(1) = -1$

в интервала [1, 2] и стъпка h = 0.1

<u>Решение :</u>

Дефинираме функциите по правилата на система *Mathematica* :

```
f[x_, y_, z_] := x + y + z^2;   g[x_, y_, z_] := (y + z)/(1. + x^2)
h = 0.1; n = 10; a = 1.; b = 2;
```

Задачата програмираме като запомняме всички решения в списък :

```
x = a; y = 1.; z = -1.;
res = {{x, y, z}};  (*Начално зареждане на списъка за резултата- res *)
For[i = 0, i < n, i++,              (* i брояч *)
   fn = f[x, y, z]; gn = g[x, y, z];
   x = x + h; y = y + h * fn; z = z + h * gn; (* Формули за пресмятане на x,y,z *)
   res = Append[res, {x, y, z, fn, gn}] (* Добавяне към края на списъка на поредните x,
  y,z,fn,gn *) ]
 TableForm[res]
```

1.	**1.**	**-1.**	**fn**	**gn**
1.1	**1.3**	**-1.**	**3.**	**0.**
1.2	**1.64**	**-0.986425**	**3.4**	**0.135747**
1.3	**2.0213**	**-0.959639**	**3.81303**	**0.267858**
1.4	**2.44552**	**-0.920172**	**4.24221**	**0.394671**
1.5	**2.91475**	**-0.86864**	**4.69224**	**0.515322**
1.6	**3.43168**	**-0.805683**	**5.16928**	**0.629572**
1.7	**3.99976**	**-0.731919**	**5.6808**	**0.737639**
1.8	**4.6233**	**-0.647913**	**6.23546**	**0.840061**
1.9	**5.30761**	**-0.554154**	**6.8431**	**0.937592**
2.	**6.05908**	**-0.451042**	**7.5147**	**1.03112**

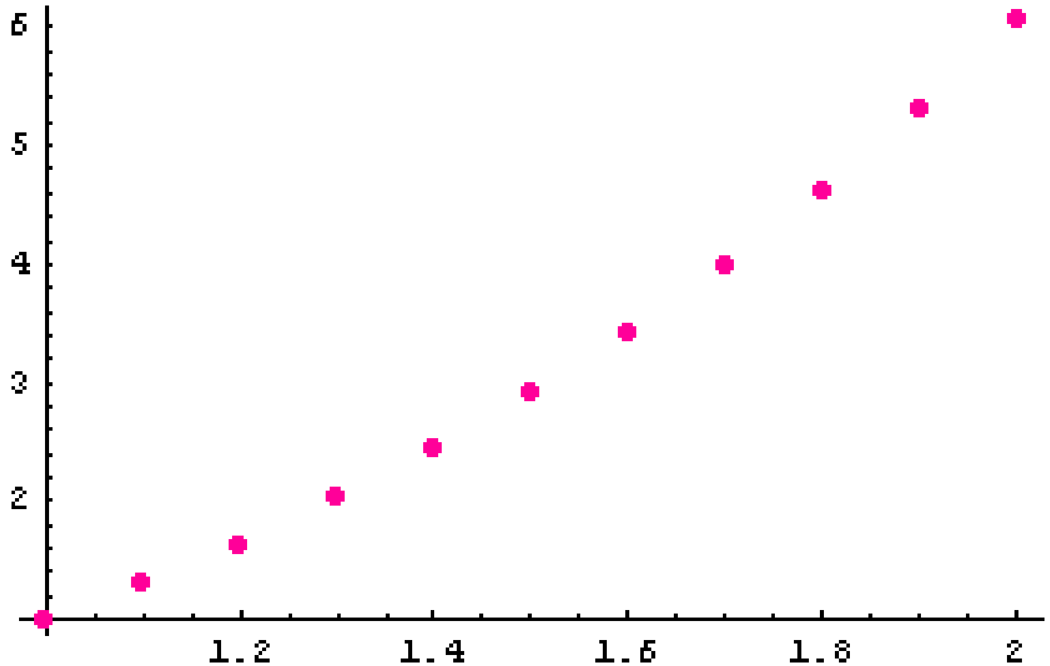
Заключение: **Решението по метода на Ойлер е във вид на таблица на неизвестните функции y(x), z(x) - съответно втора и трета колона от горната таблица. Тъй като априорната грешка на метода е O(h^2) и тук h=0.1, то тези решения трябва да се вземат закръглени само с два знака след десетичната точка.**

Графика на решението

```
py = res[[All, {1, 2}]]
(* Извличат се само първите две компоненти от всички елементи
  на списъка, т.е. {x,y}*)
gr1 = ListPlot[py, PlotStyle → {Hue[0.9], PointSize[0.02]}]
(* Чертае се точковата графика на решението y(x) *)
```

Out[142]=

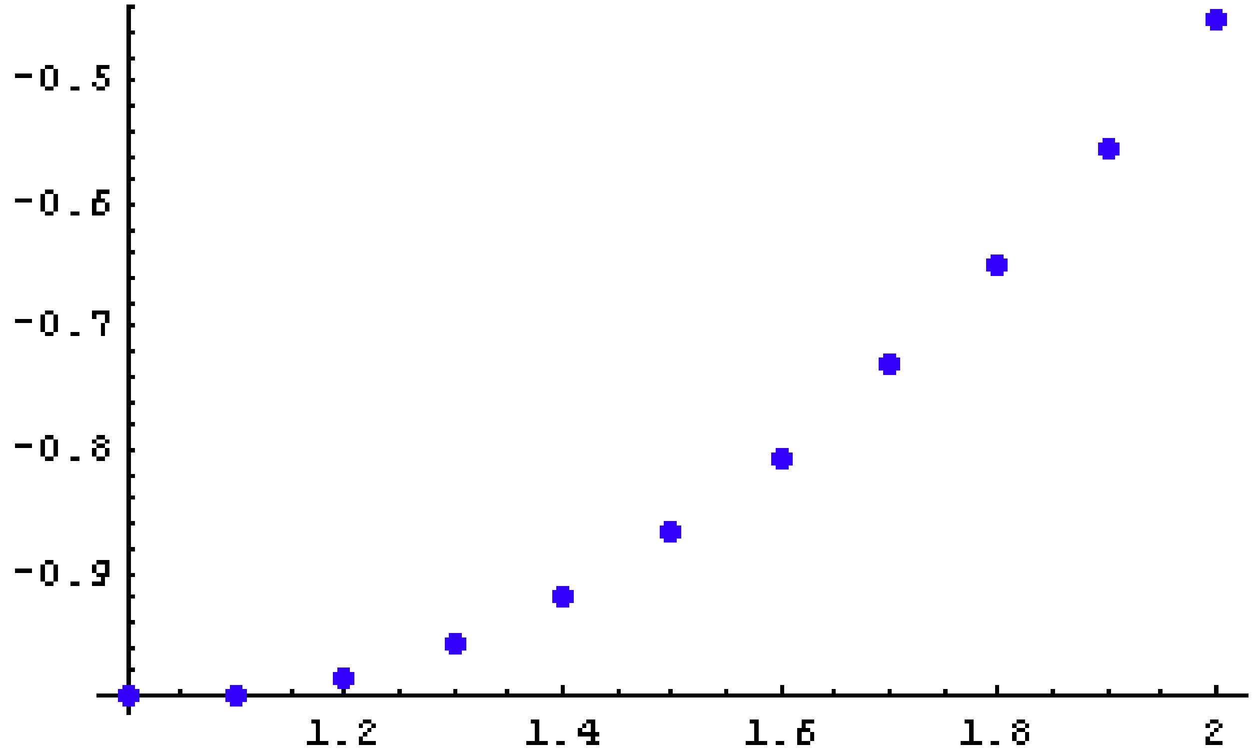
```
{{1., 1.}, {1.1, 1.3}, {1.2, 1.64}, {1.3, 2.0213}, {1.4, 2.44552},
 {1.5, 2.91475}, {1.6, 3.43168}, {1.7, 3.99976}, {1.8, 4.6233}, {1.9, 5.30761}, {2., 6.05908}}
```



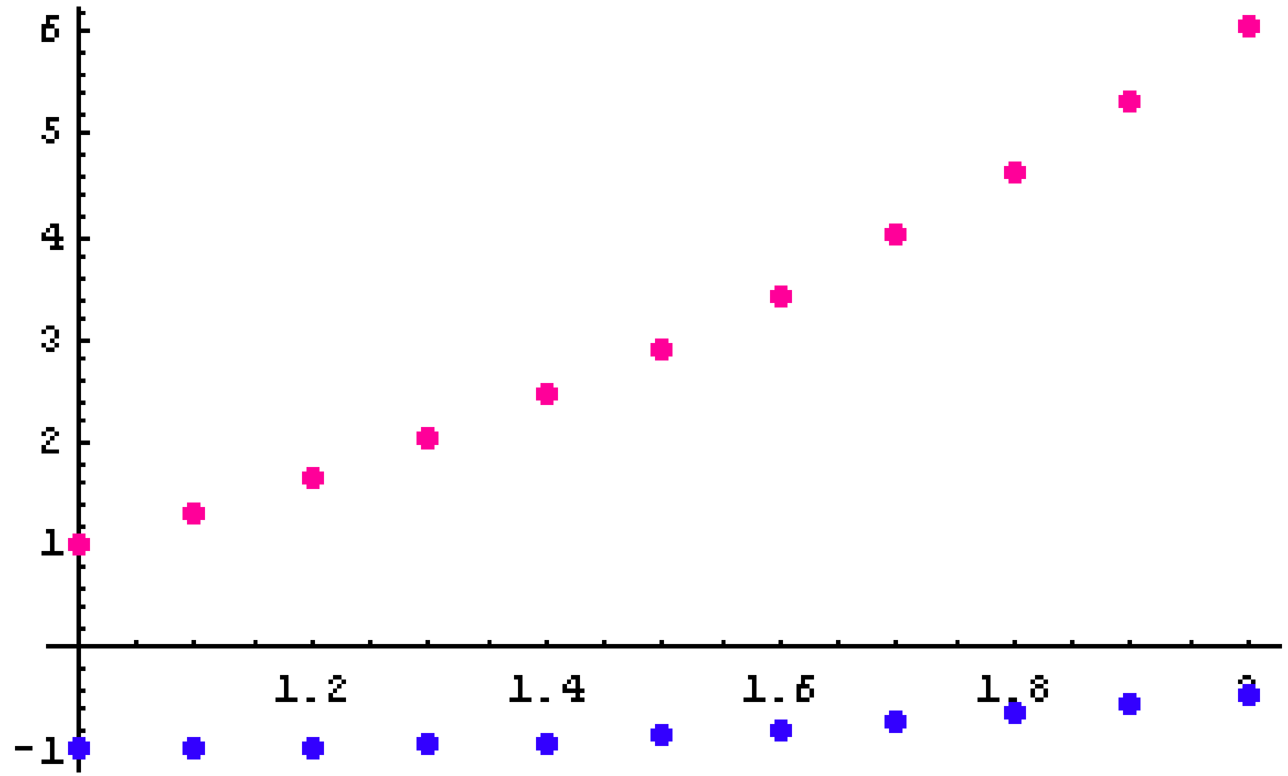
```
pz = res[[All, {1, 3}]]  (* Извличат се x и z компонентите от всички елементи на списъка *)
gr2 = ListPlot[pz, PlotStyle → {Hue[0.7], PointSize[0.02]}]
(* Чертае се точковата графика на решението z (x) *)
```

Out[143]=

{{1., -1.}, {1.1, -1.}, {1.2, -0.986425}, {1.3, -0.959639}, {1.4, -0.920172}, {1.5, -0.86864}, {1.6, -0.805683}, {1.7, -0.731919}, {1.8, -0.647913}, {1.9, -0.554154}, {2., -0.451042}}



```
Show[gr1, gr2] (* Едновременно показва двете графики за y и z*)
```



6. Модифициран метод на Ойлер – явен едностъпков метод

Решава се отново същата задача на Коши:

$$\left|\begin{array}{l} y' = f(x, y) \\ y(x_0) = y_0 \end{array}\right. , \quad x \in [x_0, b]. \qquad (2)$$

Аналогично се избират *n* точки: $x_0, x_1, \ldots, x_{n-1}, x_n = b$ в интервала $[x_0, b]$.

<u>Извод на формулата за модифицирания метод на Ойлер:</u>

Използва се междинната точка $x_{i+\frac{1}{2}} = x_i + \frac{h}{2}$. Изчисленията са на два етапа:

1) в междинната точка $x_{i+\frac{1}{2}}$ се изчислява $\boxed{y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x_i, y_i)}$ по формулата на Ойлер и функцията от $\boxed{f_{i+\frac{1}{2}} = f(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}})}$.

2) намира се търсеното $\boxed{y_{i+1} = y_i + hf_{i+\frac{1}{2}}}$. (12)

<u>Оценка на локалната грешка в точката x_i</u>. Отново развиваме функцията $y(x)$ в $x_{i+1} = x_i + h$ по формулата на Тейлър, но вече около точката $x_{i+\frac{1}{2}}$ (с половин стъпка $\frac{h}{2}$ напред от x_i). Имаме:

$$y_{i+1} = y(x_{i+1}) = y(x_{i+\frac{1}{2}} + \frac{h}{2}) = y_{i+\frac{1}{2}} + \frac{h}{2} y'_{i+\frac{1}{2}} + \frac{h^2}{4.2!} y''_{i+\frac{1}{2}} + \frac{h^3}{8.3!} y'''(\xi_1) . \qquad (13)$$

Също развиваме и с половин стъпка назад:

$$y_i = y(x_i) = y(x_{i+\frac{1}{2}} - \frac{h}{2}) = y_{i+\frac{1}{2}} - \frac{h}{2} y'_{i+\frac{1}{2}} + \frac{h^2}{4.2!} y''_{i+\frac{1}{2}} - \frac{h^3}{8.3!} y'''(\xi_2) . \qquad (14)$$

От (13) изваждаме (14) и намираме:

$$y_{i+1} - y_i = hy'_{i+\frac{1}{2}} + \frac{h^3}{8.3!}(y'''(\xi_1) + y'''(\xi_2)) = hf_{i+\frac{1}{2}} + O(h^3) . \qquad (15)$$

Като сравним с формулата на модифицирания метод на Ойлер (12) получаваме грешката между точното решение (15) и приближеното (12) във вида:

$$\left|R_{\mathrm{mod}\,Euler}(x_i)\right| \le \frac{h^3}{24} \max_{[x_i,x_{i+1}]} \left|y'''(\xi)\right| = O(h^3) \qquad (16)$$

Забележка. Модифицираният метод на Ойлер има с един порядък по-добра грешка от обикновения метод на Ойлер, като сравним (16) с (10). Това обаче е в сила, ако търсената функция притежава ограничена трета производна, което е доста по-силно изискване.

7. Метод на Ойлер-Коши – двустъпков неявен метод

Отново за същата задача на Коши

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}, \quad x \in [x_0, b]. \tag{2}$$

се избират n точки: $x_0, x_1, \ldots, x_{n-1}, x_n = b$ в интервала $[x_0, b]$. Означаваме стандартно стъпката с h.

<u>Пресмятанията се извършват на два етапа:</u>

1) $$\boxed{y_{i+1} = y_{i-1} + 2hf(x_i, y_i)} \tag{17}$$

формула предиктор – явна двустъпкова, защото използва известни значения на y с 2 стъпки назад за точките x_{i-1}, x_i;

2) $$\boxed{y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right)}, \text{ (формула на трапеца)} \tag{18}$$

или формула коректор – неявна едностъпкова, защото неизвестното y_{i+1} участва отляво и отдясно във формулата

(18).

Така за всяка точка x_{i+1} отначало се прави приближение за търсеното y_{i+1} по формула предиктор. След което един или повече пъти това y_{i+1} се уточнява чрез итериране по формулата коректор (18), например така: $\boxed{y_{i+1}^{(0)} = y_{i-1} + 2hf(x_i, y_i)}$ - нулева итерация, и 1 път итерация по формулата $\boxed{y_{i+1}^{(1)} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(0)}) \right)}$. Допуска се и евентуално още една втора итерация по формулата коректор.

Забележка. Формулите са приложими, ако в началния момент знаем поне 2 стойности, т.е. y_0 и y_1. Първото е дадено по условие, а второто можем да изчислим еднократно например с модифициран метод на Ойлер.

Оценка на локалната грешка за Ойлер-Коши в точката x_i.

Отново се използва формулата на Тейлър.

Развиваме функцията $y(x)$ в $x_{i+1} = x_i + h$ и после в $x_{i-1} = x_i - h$ около точката x_i. Имаме:

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!} y''_i + \frac{h^3}{3!} y'''(\eta_1)$$

$$y_{i-1} = y_i - hy'_i + \frac{h^2}{2!} y''_i - \frac{h^3}{3!} y'''(\eta_2) \cdot \qquad (19)$$

Като ги извадим почленно получаваме:

$$y_{i+1} - y_{i-1} = 2hy'_i + \frac{h^3}{3!} \left(y'''(\eta_1) + y'''(\eta_2) \right). \qquad (20)$$

Понеже $y(x)$ е решение на уравнението (2), то $y'_i = f_i = f(x_i, y_i)$. За локалната грешка на предиктора имаме оценка на последното събираемо

$$r_{prediktor} = y_{i+1} - y_{i-1} - 2hf_i = \frac{h^3}{3!}\left(y'''(\eta_1) + y'''(\eta_2)\right) = O(h^3), \quad \eta_1, \eta_2 \in (x_{i-1}, x_{i+1}) \quad . \tag{21}$$

За формулата коректор имаме:

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y'''(\zeta_1) \tag{22}$$

Развиваме и производната около точката x_i, т.е.

$y'_{i+1} = y'_i + hy''_i + \frac{h^2}{2!}y'''(\zeta_2)$. Оттук изразяваме втората производна

$y''_i = \frac{y'_{i+1} - y'_i}{h} + \frac{h}{2!}y'''(\zeta_2)$ и я заместваме в (22). Веднага получаваме формулата коректор, заедно с локалната грешка:

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right) + O(h^3) \quad . \tag{23}$$

Общо за целия метод

$$\left|R_{Euler-Cauchy}(x_i)\right| = O(h^3) \quad . \tag{24}$$

8. Понятие за твърди диференциални уравнения

Много често се натъкваме на ОДУ, за които разгледаните дотук, както и други методи за числено интегриране не вървят. Започва да се наблюдава рязко скачане в решението и клонене към безкрайности, т.е. – неустойчивост. По принцип това свойство е поведение на самата задача, а не се дължи на числения метод.

Такива задачи се наричат твърди (stiff equations).

Решават се общо казано по следните начини:

А) чрез използване на стандартните числени методи, но със силно намалена стъпка h, например h =0.000001, докато решението започне да се стабилизира;

Б) чрез специални числени методи, наречени обратни или неявни методи.

9. Неявен метод на Ойлер

Когато дадена задача не се решава лесно и спада към твърдите диференциални задачи, т.е. има неустойчиво решение, се прилагат различни неявни методи. Най-простият такъв метод е т.н. неявен или обратен метод на Ойлер. Той се отличава от обикновения метод на Ойлер по това, че в дясната част на рекурентната формула (11) функцията $f(x,y)$ се изчислява в точката (x_{i+1}, y_{i+1}):

$$\boxed{y(x_{i+1}) = y_i + hf(x_{i+1}, y_{i+1})}, \quad \boxed{i = 0,1,\dots,n-1}\,. \tag{25}$$

От тук се вижда, че на всяка стъпка i трябва да се решава нелинейното уравнение (25) относно неизвестната стойност на решението y_{i+1}.

Локалната грешка на неявния метод на Ойлер обаче, независимо по-добрите му изчислителни качества (устойчивост на изчислителния процес) е от същия порядък, както обикновения метод на Ойлер, т.е. $O(h^2)$ - локална грешка на метода и $O(h)$ - глобална грешка.