פרוייקט: 1B

פונקציות:

$$f(x)=\frac{\sin(x)}{\sinh(x)}$$

$$g(x)=\sin(\tan(\cos(x)))$$

תחום:

$$x\in[-2,3]$$

תנאי התחלה:

$$y(x_0)=y_0=7$$

מטרת הפרויקט:

- לפתח שיטה נומרית – שיטת Euler לחישוב משוואה דיפרנציאלית
- לפתור משוואה $y^I(x)=f(x)y+g(x)$ בין הגבולות מ- 2- עד 5 לפי שיטה הנ"ל עם תנאי התחלה $y(-2)=7$
- לכתוב תוכנה בשפת C++ לפיתרון משוואה דיפרנציאלית
- לבצע בדיקה תוכנה

מונחים בסיסיים.

שינוי:

אם משתנה x מקבלת ערך $x=x_1$ ואחרכך ערך $x=x_2$ לכן הפרש x_2-x_1 נקרא **שינוי** של ערך x. שינוי יחול להיות חיובי, שלילי או שווה לאפס. מילה שינוי מוגדרת כי Δ. Δx (קוראים "דלטה אקס") מגדיר שינוי ערך של x כך ש:

$$\boxed{\Delta x = x_2 - x_1}$$

<u>נגזרת:</u>

נניח $y=f(x)$ זאת פונקציה רציפה בקטע (a,b) ו-x זאת נקודה בתוך הקטע.
נשנה את ה-x במרחק מאוד קטן $\Delta x\to 0$, פונקציה $y=f(x)$ גם תישתנה ב-Δy כךש: $\Delta y = f(x+\Delta x)-f(x)$, כאשר $\Delta x\to 0$ אז שינוי $\Delta y\to 0$
גבול שאליו שואף יחס $\frac{\Delta y}{\Delta x}$ כאשר $\Delta x\to 0$ ז.א. $\lim\limits_{\Delta x\to 0}\frac{f(x+\Delta x)-f(x)}{\Delta x}$ בעצמו פונקציה של ארגומנת x. הפונקציה הזאת נקראת **נגזרת** של פונקציה $f(x)$ וסימונה $f^{I}(x)$ או y^{I}.

<u>דיפרנציאל:</u>

נניח דלתה של פונקציה $y=f(x)$ מחולק לשני חלקים $\Delta y = A\Delta x+\alpha$, איפה ש-$A$ לא תלוי מ-Δx (ז.א. קבוע עבור ערך הנתון של ארגומנט x) ו- $\lim\frac{\Delta x}{\alpha}\to 0$.
לכן חלק ראשון (ראשי) פרופורציאנלי ל-Δx נקרא **דיפרנציאל** של פונקציה $f(x)$ וסימונו
$df(x)$.

<u>טאורמה של ניוטון-לייבניץ:</u>

אינטגרל מדיפרנציאל של פונקציה $F(x)$ שווה לשינוי פונקציה בקטע אינטגרציה

$$\boxed{\int_a^b dF(x) = F(b) - F(a)}$$

משוואה דיפרנציאלית

נקראת משוואה שמכילה ניגזרות של פונקציה לא ידוע (או קמה פונקציות לא ידועות). במקום נגזרות יכולות להיות דיפרנציאלים. אם פונקציות לא ידועות תלויות ממשתנה יחיד אז משוואות דיפרנציאליות נקראות **רגילות**.

אם פונקציות לא ידועות תלויות מקמה משתנים אז משוואות דיפרנציאליות נקראות **חילקיות**. נראות של משוואה דיפרנציאלית רגילה:

$$\boxed{F(x, y, y^{I}, y^{II}, \ldots, y^{(n)}) = 0}$$

סדר של משוואה דיפרנציאלית נקרא סדר של נגזרת הגדולה במשוואה.

פונקציה $y = \varphi(x)$ נקראת **פיתרון** של משוואה דיפרנציאלית אם לאחר הצגתה למשוואה מתקיים השוויון.

מטרה: למצאה כל פיתרונות של משוואה הנתונה.

במקרים פשותים צריכים למצוא תוצאות של אינטגרל, לכן פיתרון של משוואה דיפרנציאלית נקראה גם **אינטגרל**, ותהליך חיפוש של כל הפיתרונות – **אינטגרציה** של משוואה דיפרנציאלית.

משוואה דיפרנציאלית מסדר ראשון נראת $F(x, y, y^{I}) = 0$ ופיתרון שלה יחסית ל- y^{I} נראה $y^{I} = f(x, y)$

מניחים שפונקציה $f(x, y)$ מוגדרת חד-משמעי ורציפו בקטע מסויימת, מחפשים אינטגרלים בקטע. למשוואה דיפרנציאלית מסדר ראשון יש אין סוף פיתרונות.

פיתרון המתאים של משוואה נקראה **פיתרון חילקי**.

אוסף של כל פיתרונות חילקיות נקראה **פיתרון כללי**.

אותו רוצים להציג כפונקציה $y = \varphi(x, C)$, (כאשר C-קבוע) שנותנת כל פיתרון חילקי (עבור בחירה של C מסויים).

לפעמים הצגה הזאת בילתי אפשרית אפילו טאורטית.

אבל פיתרון חילקי שעובר דרך נקודה מסויימת (x_0, y_0) ניתן למצוא אם לא במדוייק דרך פונקציות אלמנטריות אז בקירוב עם דיוק כלשהו.

מספרים x_0 ו- y_0 נקראות ***תנאי התחלה***.

דרישות והגבלות,

- לבנות מודל לפיתרון של משוואה דיפרנציאלית מסוג $y^I = f(x)y + g(x)$ בקטע (a,b) עם תנאי התחלה y_0.
- להוכיח ששיטה מחפשת פיתרון בצורה מדויקת
- חישוב מתבצעה זמן קצר
- שיטה אמינה
- לכתוב תוכנה קלה ונוכה לשימוש

קלט של תוכנה:

- הגדרת גבולות (שמאל וימין)
- הגדרת דיוק של חישוב
- הכנסת תנאי התחלה
- הכנסת נוסחה של פונקציה לחישוב אינטגרל $f(x)$
- הכנסת נוסחה של פונקציה לחישוב אינטגרל $g(x)$

פלט של תוכנה:

- אם סיום המוצלח של חישוב להדפיס תוצאה והודאה המתאימה
- אם סיום לא מוצלח של חישוב להדפיס הודאה המתאימה

<u>שיטת Euler</u>

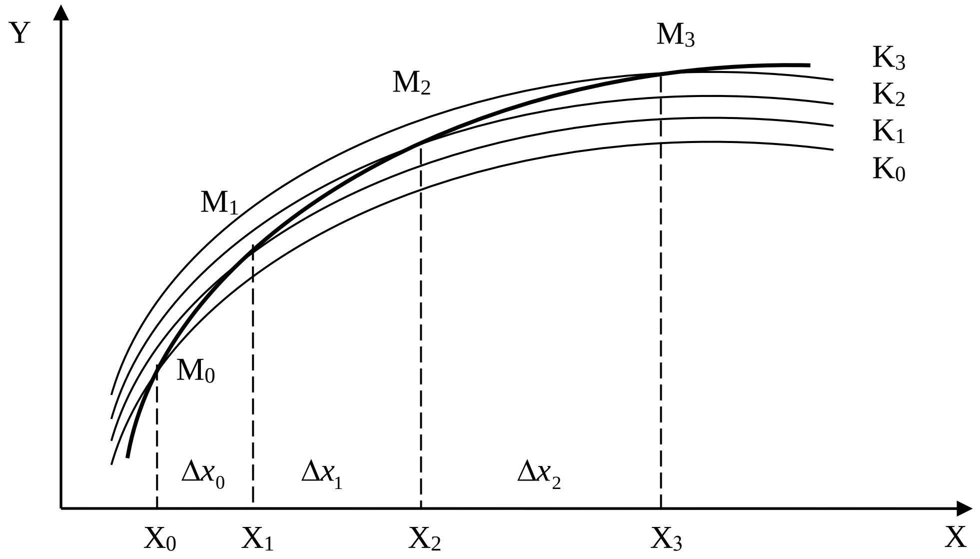
נשתמש בטאורמה ניוטון-לייבניץ:

$$\int_a^b dF(x) = \int_a^b y^I(x)dx = F(b) - F(a)$$

נניח נתונה משוואה $y^I = f(x,y)$ ותנאי התחלה $x = a, y(a) = y_0$.

רוצים למצוא פיתרון של משוואה בקטע מסויימת (a,b).

נחלק קטע ל- n חלקים (שוים או לא שוים) בעזרת נקודות $x_1, x_2, \dots, x_{n-1}$ כמו בציור:



בקטע (x_0, x_1) מניחים כי $y = y_0 + f(x_0, y_0)(x - x_0)$

ז.א. במקום קו אינטגרלי M_0K_0 שהוא מדוייק לוקחים משיק שלו M_0M_1.

בנקודה $x = x_0$ מקבלים קירוב של פיתרון שמחפשים

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0) = y_0 + f(x_0, y_0)\Delta x_0$$

בקטע (x_1, x_2) מניחים כי $y = y_1 + f(x_1, y_1)(x - x_1)$

ז.א. במקום קו האינטגרלי M_0K_0 לוקחים משיק שלו M_1M_2

אם נמשיך באותו תהליך נקבל שאר של קירובים:

$$y_2 = y_1 + f(x_1, y_1)\Delta x_1$$
$$y_3 = y_2 + f(x_2, y_2)\Delta x_2$$

....

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})\Delta x_{n-1}$$

ברך כלל אדיף לחלק קטע לחלקים שווים.
במקרא שידעות נגזרות של פונקציה עד סדר מסויים k אז נוסחה נראת כך:

$$y(b) = y_n = y_{n-1} + \Delta x\left(f(x_{n-1}, y_{n-1}) + \frac{f^{I}(x_{n-1}, y_{n-1})}{2}\Delta x + \ldots + \frac{f^{(k)}(x_{n-1}, y_{n-1})}{(k+1)!}(\Delta x)^k \right)$$

כדי להרים דיוק צריכים או לחלק קטע למספר גדול של חלקים או להוסיף ניגזרת הבאה של פונקציה.
בדרך כלל לוקחים עד נגזרת שלישית.
פיתרון למשוואה יהיה ביטוי:

$$\boxed{y(b) - y(a)}$$

קוד של תוכנה

```
#include <iostream.h>
#include <math.h>
#include <assert.h>
/*
	dlF ->  given F(x,y,y') y'=f(x)*y + g(x)
	dlD ->  F'(x,y,y')
	dlD2 -> F"(x,y,y')
	dlT -> T(ti , wi) = F(ti,wi) + (h/2)F'(ti,wi) + (h^2/6)*F"(ti,wi)
*/
double dlF(double , double );
double dlD(double , double );
double dlD2(double , double );
double dlT(double , double , double);


void main()
{
	/*
		dlw[0] -> wi-1
		dlw[1] -> wi
		dlh -> h
		dla -> a
		dlb -> b
		dly0 -> Yo
		dlOld -> answer when N = 2^(i-1)
		dlCurrent -> answer when N = 2^i
		dlEps -> epsilon
	*/
	double dlw[2] , dlt =0 , dlh , dla , dlb , dly0 , dlOld , dlCurrent = 100000 , dlEps;
	int iN = 0 , i= 0;
	cout<<"insert lowwer border : ";
	cin>>dla;
	cout<<"insert upper border : ";
	cin>>dlb;
	assert(dlb > dla);
	cout<<"insert initial condition for Yo : ";
	cin>>dly0;
	cout<<"insert epsilon : ";
	cin>>dlEps;
	assert (dlEps > 0);
	do
	{
		dlOld = dlCurrent;
		iN = pow(2 , i);
		dlh = (dlb - dla)/iN;
		dlt = dla;
		dlw[0] = dly0;
		do
		{
		dlt += dlh;
		dlw[1] = dlw[0] + dlh*dlT(dlt , dlw[0] , dlh);
		dlw[0] = dlw[1];
		}
```

```
			while(dlt < dlb);
			dlCurrent = dlw[0] - dly0;
			i++;
		}
		while(fabs(dlCurrent - dlOld) > dlEps);

cout<<"with "<<iN<<" steps"<<endl;
   cout.precision(15);
		cout<<"y(b) - y(a) = "<< dlCurrent<<endl;

}
double dlT(double _dlt, double _dlw , double _dlh)
{
		return (
				dlF(_dlt , _dlw) + (_dlh/2)*dlD(_dlt , _dlw) + (pow(_dlh,2)/6)*dlD2(_dlt ,
_dlw)
				);
}
double dlF(double _dlt, double _dlw)
{
		return (
				(sin(_dlt)*_dlw/sinh(_dlt))+
				 sin(tan(cos(_dlt)))
				);
}
double dlD(double _dlt, double _dlw)
{
		return (
				(cos(_dlt)*sinh(_dlt) - sin(_dlt)*cosh(_dlt))*_dlw / pow(sinh(_dlt),2) +
				 sin(_dlt)*dlF(_dlt,_dlw)/sinh(_dlt) +
				(-cos(tan(cos(_dlt))))*(1 + pow(tan(cos(_dlt)),2))*sin(_dlt)

				);
}
double dlD2(double _dlt, double _dlw)
{
		return (
				( ( -2 )* (sin(_dlt)/sinh(_dlt))-
				2*cos(_dlt)*cosh(_dlt)/sinh(_dlt) +
				2*sin(_dlt)*pow(cosh(_dlt),2)/pow(sinh(_dlt),3))*_dlw +
			2* (cos(_dlt)*sinh(_dlt) - sin(_dlt)*cosh(_dlt))*dlF(_dlt,_dlw)/pow(sinh(_dlt),2)+
				sin(_dlt)*dlD(_dlt,_dlw)/sinh(_dlt) +
				pow( - sin(tan(cos(_dlt)))*(1 + pow(tan(cos(_dlt)),2)),2)*pow(sin(_dlt),2) +
		   2*cos(tan(cos(_dlt))) * tan(cos(_dlt))*(1 + pow(tan(cos(_dlt)),2))*pow(sin(_dlt),2) -
				 cos(tan(cos(_dlt)))*(1 + pow(tan(cos(_dlt)),2))*cos(_dlt)
				);
}
```

<u>תוצאות הרצה של תוכנה.</u>

```
insert lowwer border : -2
insert upper border : 3
insert initial condition for Yo : 7
insert epsilon : 0.1
with 32768 steps
y(b) - y(a) = 123.434310090823
Press any key to continue
```

```
insert lowwer border : -2
insert upper border : 3
insert initial condition for Yo : 7
insert epsilon : 0.01
with 524288 steps
y(b) - y(a) = 123.262653784169
Press any key to continue
```

```
insert lowwer border : -2
insert upper border : 3
insert initial condition for Yo : 7
insert epsilon : 0.001
with 2097152 steps
y(b) - y(a) = 123.254070942225
Press any key to continue
```

בדיקות תוכנה

ניקח משוואה דיפרנציאלית $2ydx+xdy=0$ בקטע סגור $[1,2]$, עם תנאי התחלה $(x_0=1\Rightarrow y_0=1)$. למשוואה הדאת יש פיתרון אנליטי:

$$2ydx=-xdy\Rightarrow\frac{dy}{y}=-2\frac{dx}{x}\Rightarrow\ln y=-2\ln x\ln C\Rightarrow y=\frac{C}{x^2}\Rightarrow x^2y=C$$

נציב תנאי התחלה ונקבל ש-C שווה ל-1.

כיבלנו פיתרון פרטי של משוואה דיפרנציאלית הנתונה: $\boxed{y=x^{-2}}$

נגדיר פונקציה $f(x)=y^I=-2\frac{y}{x}$ ונחשב ניגזרות שלה עלמנת להציב בתוכנה.

$$f^I(x)=-2\frac{y^Ix-y}{x^2}=-\frac{4y+2y}{x^2}=6\frac{y}{x^2}$$

$$f^{II}(x)=6\frac{y^Ix^2-2xy}{x^4}=6\frac{-2y-2y}{x^3}=-24\frac{y}{x^3}$$

לפי טאורמה של נייטון-לייבניץ $y(b)-y(a)=y(2)-y(1)=0.25-1=-0.75$

ע"ס נתונים האלה פלט של תוכנית יהיה:

א) נחשב נקודות לפי נוסחה $y_n=y_{n-1}+hf(x)$

שגיאה	תוצאה	dx	מספר קטעים
1.32×10^{-2}	-.7368421053	10^{-1}	10^1
1.26×10^{-3}	-.7487437186	10^{-2}	10^2
1.25×10^{-4}	-.7498749375	10^{-3}	10^3
1.25×10^{-5}	-.7499874994	10^{-4}	10^4
1.25×10^{-6}	-.7499987500	10^{-5}	10^5
1.25×10^{-7}	-.7499998750	10^{-6}	10^6
1.26×10^{-8}	-.7499999874	10^{-7}	10^7

ב) נחשב נקודות לפי נוסחה $y_n=y_{n-1}+h\left(f(x)+\frac{h}{2}f^I(x)\right)$

שגיאה	תוצאה	dx	מספר קטעים
5.66×10^{-2}	-.6933715729	10^{-1}	10^1
5.06×10^{-3}	-.7449370958	10^{-2}	10^2
5.01×10^{-4}	-.7494993746	10^{-3}	10^3
5.00×10^{-5}	-.7499499937	10^{-4}	10^4
5.00×10^{-6}	-.7499949999	10^{-5}	10^5
5.00×10^{-7}	-.7499995000	10^{-6}	10^6
5.01×10^{-8}	-.7499999499	10^{-7}	10^7

ניקח אוד דוגמא אחד.

ניקח משוואה דיפרנציאלית $y^I=\frac{1}{2}xy$ בקטע סגור $[0,1]$, עם תנאי התחלה $(x_0=0\Rightarrow y_0=1)$.

למשוואה הדאת יש פיתרון אנליטי:

$$\frac{dy}{y}=\frac{1}{2}xdx \Rightarrow \int\frac{dy}{y}=\frac{1}{2}\int xdx \Rightarrow y=Ce^{\frac{1}{4}x^2}$$

נציב תנאי התחלה ונקבל ש-C שווה ל-1.

כיבלנו פיתרון פרטי של משוואה דיפרנציאלית הנתונה: $\boxed{y=e^{\frac{1}{4}x^2}}$

נגדיר פונקציה $f(x)=y^I=\frac{1}{2}xy$ ונחשב ניגזרות שלה עלמנת להציב בתוכנה.

$$f^I(x)=\frac{1}{2}xy^I+\frac{1}{2}y=\frac{1}{4}x^2y+\frac{1}{2}y$$

$$f^{II}(x)=\frac{1}{2}xy+\frac{1}{4}x^2y^I+\frac{1}{2}y^I=\frac{1}{2}xy+\frac{1}{4}xy+\frac{1}{8}x^3y=0.75xy+0.125x^3y$$

לפי טאורמה של נייטון-לייבניץ $y(b)-y(a)=y(1)-y(0)=e^{0.25}-1=0.2840254167$

ע"ס נתונים האלה פלט של תוכנית יהיה:

א) נחשב נקודות לפי נוסחה $y_n=y_{n-1}+hf(x)$

שגיאה	תוצאה	dx	מספר קטעים
3.61×10^{-2}	.2479716487	10^{-1}	10^1
3.73×10^{-3}	.2802950772	10^{-2}	10^2
3.74×10^{-4}	.2836510574	10^{-3}	10^3
3.74×10^{-5}	.2839879674	10^{-4}	10^4
3.75×10^{-6}	.2840216716	10^{-5}	10^5
3.75×10^{-7}	.2840250422	10^{-6}	10^6
3.75×10^{-8}	.2840253792	10^{-7}	10^7

ב) נחשב נקודות לפי נוסחה $y_n=y_{n-1}+h\left(f(x)+\frac{h}{2}f^I(x)\right)$

שגיאה	תוצאה	dx	מספר קטעים
4.00×10^{-3}	.2800238019	10^{-1}	10^1
4.07×10^{-5}	.2839846919	10^{-2}	10^2
4.08×10^{-7}	.2840250088	10^{-3}	10^3
4.10×10^{-9}	.2840254126	10^{-4}	10^4
1.00×10^{-10}	.2840254166	10^{-5}	10^5
1.00×10^{-10}	.2840254166	10^{-6}	10^6
1.00×10^{-10}	.2840254166	10^{-7}	10^7

נשנה גבולות של קטע:

ניקח $dx=10^{-4}$ ותוצאת חישוב 0.2840254126, נשנה גבולות ונבוק שינוים:

שינוי y	תוצאה	b	a
3.24×10^{-3}	.2807817875	1	-0.1
3.24×10^{-3}	.2807828651	1	0.1
5.96×10^{-2}	.2244319433	0.9	0
6.92×10^{-2}	.3531884880	1.1	0

<u>מסקנה:</u> שינוי קטן של גבולות לא גורם לשינוי גדול של פיתרון.

נריץ תוכנה עבור משוואה $y^I = \sin(\sin^2(x))y + e^x \cos(x^2)$

א) נחשב נקודות לפי נוסחה $y_n = y_{n-1} + hf(x)$

תוצאה	dx	מספר קטעים
81.9099088427	10^{-1}	10^1
96.1455977841	10^{-2}	10^2
97.6727965135	10^{-3}	10^3
97.8266272500	10^{-4}	10^4
97.8420215123	10^{-5}	10^5
97.8435610504	10^{-6}	10^6
97.8437150110	10^{-7}	10^7

ב) נחשב נקודות לפי נוסחה $y_n = y_{n-1} + h\left(f(x) + \frac{h}{2} f^I(x)\right)$

תוצאה	dx	מספר קטעים
83.5619240979	10^{-1}	10^1
96.3447258526	10^{-2}	10^2
97.6930836490	10^{-3}	10^3
97.8286597451	10^{-4}	10^4
97.8422247996	10^{-5}	10^5
97.8435813795	10^{-6}	10^6
97.8437170439	10^{-7}	10^7

<u>מסקנות:</u>

- **מספר רב של איטרציות גורם לעלות זמן חיפוש של פיתרון.**
- **כדי להרים דיוק צריכים:**

א) להגדיל מספר איטרציות (לקחת יותר נקודותביינים)
ב) לחשב ביטוי של ניגזרת נוספת (מסדרים הבאים)

סיפרות, תוכנות עזר.

1. Calculus with analytic geometry / C. H. Edwards , Jr., David E. Penney --- 5^{th} edition.

2. Guide of higher mathematics / M. Vigorodsky.

3. Numerical Recipes in C/ William H. Press, Saul A. Teukolsky, William T. Vetterling,
   Brian P. Flannery --- The Art of Scientific Computing 2^{nd} Edition