Całkowanie numeryczne układów równań różniczkowych zwyczajnych.

Laboratorium 5

Wykorzystanie w praktyce metody Eulera oraz Rungego - Kutty 4 rzędu.

Jakub Mroczkowski gr.9

Numer indeksu: 304336

Data wykonania:12.05.2020r.

Użytą funkcję w programie wykorzystałem ze strony naszego wydziału (rk4).

1.Cel zajęć.

Celem laboratorium numer 5 z przedmiotu Informatyka 2, było zapoznanie z praktycznym wykorzystaniem metody Eulera oraz R-K4 do rozwiązania przykładowych równań różniczkowych zwyczajnych w naszym przypadku równań opisujących ruch wahadła matematycznego.

$$\begin{cases} \frac{d^2\alpha}{dt^2} = -\frac{g}{l}\sin(\alpha) \\ \alpha(t_0) = \alpha_0 \\ \frac{d\alpha}{dt}(t_0) = \omega_0 \end{cases}$$

W zadaniu mieliśmy przedstawiony następujący układ równań. Po przekształceniach oraz odpowiednim podstawieniu otrzymaliśmy finalny układ równań.

$$\begin{cases} \frac{d\omega}{dt} = F_1(\alpha, \omega, t) \\ \frac{d\alpha}{dt} = F_2(\alpha, \omega, t) \end{cases}$$

Szukanymi funkcjami są oczywiście wychylenie oraz prędkość kątowa wahadła, dodatkowo w poleceniu z ćwiczenia poproszono nas o obliczenie energii całkowitej w ruchu wahadła. Jako, że w raporcie numer 4 pisałem już o metodach Eulera oraz R-K4 , wykonałem tylko ćwiczenie do tego dołączę odpowiednie wykresy , wraz z wnioskami.

2. Kod wraz z komentarzem.

```
void veuler(double t, double *X, double h, int n,void (* fun)(double, double *,double *), double *X1);
void rhs_fun(double t, double *X,double *F);
int main()


{
    FILE*fp;
    fp=fopen("cw.txt","w");
    int n =2;
    double t,tk,h,a0,w0;
    double *Xe,*X1e , *Xr,*Xr1;
    Xe=(double*)malloc(n*sizeof(double));
    X1e=(double*)malloc(n*sizeof(double));
    Xr=(double*)malloc(n*sizeof(double));
    Xr1=(double*)malloc(n*sizeof(double));
    tk=10.0;
    double g,l;
    g = 9.81;
    l=1.0;
    h=0.1;
    w0=0.0;
    a0=0.4;
```

Mój kod rozpoczyna się od nagłówków funkcji, które musiałem wprowadzić.

Są to odpowiednio veuler i rhs_fun , nazwy takie jak poproszono w poleceniu.

Analizując linijka po linijce kod zaczynam od stworzeniu pliku, do którego będę mógł zapisać wyniki, co ułatwi mi stworzenie wykresów. Następnie otwieram plik w trybie write. Wprowadzam zmienne czasowe t,tk. Krok całkowania h, oraz a0 i w0 czyli wychylenie oraz prędkość kątową. Deklaruję 4 tablice w wymiarze n =2 , Xe, Xr wprost z polecenia ,a także X1e oraz Xr1 jako tablice do przechowania wartości zmiennych zależnych w kroku t+h, również wprost z polecenia. Wartości t z przedziału <0,10>. Deklaruję następnie g jako 9.81 , l jako 1 i wartość h jako 0.1 na powyższym zdjęciu aczkolwiek do wykresów używałem kroku 0.01.

```
double m = 5;
Xe[0] = w0;
Xe[1] = a0;
Xr[0]=  w0;
Xr[1] = a0;

double Em,Em1;
printf("Energia calkowita euler\t\tEnergia calkowita RK-4\n");
for(t=0;t<=tk; t = t+h)

{

    veuler(t,Xe,h,n,rhs_fun,X1e);
    vrk4(t,Xr,h,n,rhs_fun,Xr1);
    Em = (m*l*l/2*Xe[0]*Xe[0])+(m*g*l*(1-cos(Xe[1])));
    Em1 = (m*l*l/2*Xr[0]*Xr[0])+(m*g*l*(1-cos(Xr[1])));
    printf("%lf\t\t\t%lf\n",Em,Em1);
    fprintf(fp, "%lf\n" ,t);
    Xe[0] = X1e[0];
    Xe[1] = X1e[1];
    Xr[0] = Xr1[0];
    Xr[1] = Xr1[1];


}
system("PAUSE");
```

Następnym krokiem jest przyjęcie odpowiednich wartości przez moje tablice dla metody Eulera pierwsze miejsce w tablicy to prędkość, drugie to wychylenie. Analogicznie RK-4. Tworzę następującą pętlę tym razem w przeciwieństwie to poprzedniego kodu od razu w pętli operuję na zmiennej niezależnej wartości t w pętli muszą być mniejsze od tk co wynika również ze wzoru H = tk-tp/n. Oczywiście zwiększam co iteracje zmienną niezależną o krok całkowania.

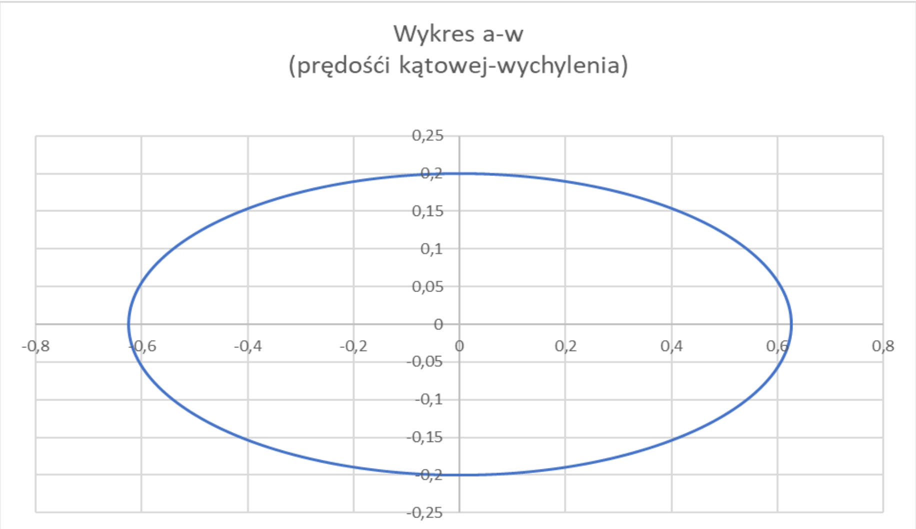
Używam funkcji, które zaraz opiszę. Do wzorów na energię całkowitą podstawiam bezpośrednio wyniki dwóch metod. Następnie przyrównuje odpowiednie tablice do siebie tak, żeby zapisać wartość do tablicy (X1-w poleceniu zadania) u mnie Xr1 oraz X1e.

```
void rhs_fun(double t, double *X,double *F)

{
    double g,l;
    g = 9.81;
    l = 1.0;
    F[0] = (-1.0)*g/l*sin(X[1]);
    F[1] = X[0];
}
void veuler(double t, double *X, double h, int n,void (* fun)(double, double *,double *), double *X1)

{
    double *F;
    F = (double*)malloc(n*sizeof(double));
    fun(t,X,F);
    X1[0] = X[0] + (h*F[0]);
    X1[1] = X[1] + (h*F[1]);
    free(F);


}
```

Na samym końcu kodu znajduje się funkcja rhs_fun, którą wykonałem zgodnie z poleceniem, ma ona obliczać prawe strony równania, na pochodną prędkości kątowej oczywiście mamy wzór natomiast na pochodna wychylenia jest równa prędkości kątowej. Druga funkcją jest veuler nazwa jak w poleceniu przyjmuje ona wskaźnik do funkcji obliczającej prawe wartości i oczywiście takie zmienne jak krok całkowania, zmienna niezależną tablice z wartościami, a w środku wywołanie funkcji fun i oczywiście obliczenie ze wzoru Eulera dwóch równań różniczkowych.

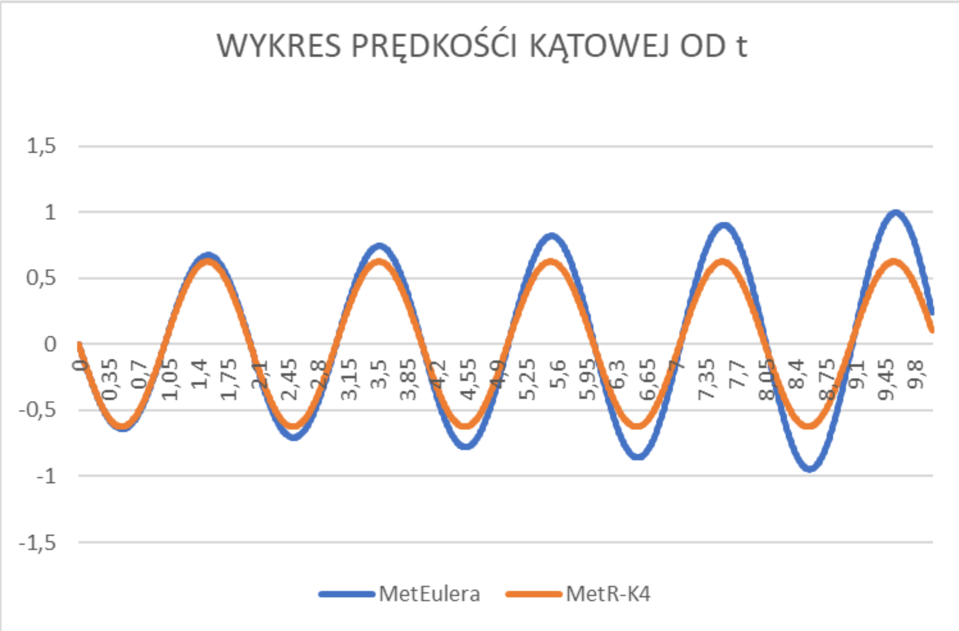
3. WYKRESY I WNIOSKI.

W zadaniu poproszono, aby zrobić wykres trajektorii (a-w).

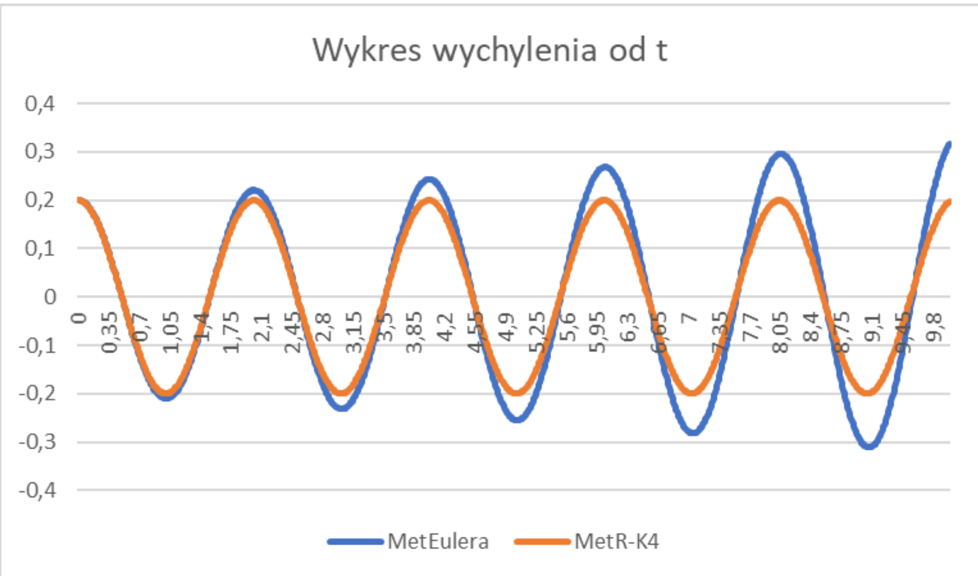


Co ciekawe wyszła piękna elipsa.

Następnie wykonałem wykresy prędkości kątowej oraz wychylenia, aby zobaczyć czy wyjdą odpowiednie wykresy, a moje obliczenia pokrywają się z rzeczywistością. Wykresy zrobiłem dla a = 0.2, warto dodać, że zapisywałem po kolei każdą kolumnę danych dlatego w moim kodzie linijka z fprintf zapisuje tylko t na zdjęciu, aczkolwiek po kolei zapisywałem w notatniku każda kolumnę danych osobno po kolei odpalając program dla innych danych przy printf.



Pierwszy wykres przedstawia sinusoidalny przebieg prędkości kątowej. Metoda eulera ma większą amplitudę od metody R-K4 co oczywiście podkreśla mniejszą dokładność metody Eulera, dla kroku całkowania 0.01. Jeżeli zwiększam krok całkowania to łatwo zauważam, że obie metody osiągają podobne rezultaty.



Kolejnym wykresem jest wychylenie w funkcji t. Minimalnie różnica pomiędzy amplitudami w obu metodach jest większa niż na wykresie prędkości, aczkolwiek wykres sugeruje również, że moje wyniki powinny być poprawne, po raz kolejny metoda Eulera jest wyraźnie mniej dokłada od metody R-K4.

Następnie obliczyłem również energię całkowitą dla obu metod.

4.634792	3.871860
5.071339	3.871814
5.555508	3.871775
6.095837	3.871732
6.693357	3.871673
7.340102	3.871610
8.027698	3.871559
8.757613	3.871522
9.540341	3.871487
10.385158	3.871442
11.296252	3.871390
12.283910	3.871342
13.378927	3.871304
14.626483	3.871261
16.049153	3.871205
17.608989	3.871141
19.224014	3.871088
20.840197	3.871050
22.474039	3.871015
24.173382	3.870971
25.960360	3.870920
27.835501	3.870871
29.840392	3.870832
32.116631	3.870791
34.875122	3.870737
38.213575	3.870673
41.894632	3.870617
45.417216	3.870577
48.444957	3.870543
51.048455	3.870501

Lewa kolumna opisuje metodę Eulera, gdzie widać jak bardzo w czasie rośnie dana energia, wynika to po pierwsze z faktu, że krok całkowania przyjąłem, dla tych obliczeń 0.1 tutaj obliczenie energii wyraźnie ukazało mi jak istotna w metodzie Eulera jest mała wartość kroku całkowania, dla metody R-K4 rozbieżności są idealnie małe.

4. Podsumowanie.

Wykonując to laboratorium na pewno nabrałem ogólniejszego pojęcia o zastosowaniu metod numerycznych do rozwiązywania równań różniczkowych zwyczajnych, co ułatwił mi przykład fizyczny, który jest ciekawy oraz dobrze testuje obie metody. Po raz kolejny można wywnioskować empirycznie, że metoda Eulera jest mniej dokładna, a metoda R-K4 zarysowuje wyraźnie swój 4-rząd zbieżności wygrywając z metodą Eulera, która jak wszystko ma też swoje zalety jest prostsza w zastosowaniu i napisaniu, więc orientacyjnie obliczając równania można używać tej metody, ale trzeba pamiętać , że krok całkowania musi być odpowiednio mały.