
Modelowanie komputerowe

Raport

Porównanie metod rk2 (Heuna i MidPoint), rk4 i rk45
Fehlberga

Katarzyna Gajewska

Michał Kreft

Tomasz Biel

29 listopad 2021

1. Wstęp

Jako przedmiot naszego projektu wybraliśmy temat będący zadaniem polegającym na porównaniu metod rk2, rk4 oraz rk45 na podstawie równania ruchu oscylatora harmonicznego i dwóch zadanych równań. Nie będziemy skupiać się na omawianiu analitycznym poszczególnych metod - zrobiliśmy już to w prezentacji. Skupimy się w głównej mierze na naszej wykonanej pracy, czyli przedyskutujemy nasze ćwiczenie omawiając wyniki oraz kod.

2. Metody Rungego-Kuty i ich zastosowanie

Zbiór metod różnego rzędu, które pozwalają na całkowanie równań różniczkowych zwyczajnych przeprowadzając je przez szereg obliczeń krok po kroku. Dokładność całkowania jest zdeterminowana poprzez wielkość kroku jaki wybierzemy. W metodach tych nachylenie szacowane jest na podstawie kilku punktów wewnątrz przedziału. Różne metody RK klasyfikowane są ze względu na ich rząd (odpowiadający liczbie punktów wziętych do szacowania nachylenia). Sens używania tych metod w całości mieści się w fachu programistycznym z powodu na zdecydowaną przewagę maszyn nad człowiekiem w szybkości obliczeń.

3. Analiza ćwiczenia

Na samym początku zaimplementowaliśmy metody Rungego-Kuty podając do tego funkcję, która ulegnie całkowaniu. Nasze wyniki porównaliśmy z rozwiązaniem metodą *odeint* przyjmując ją jako wartość dokładną. Jedynie w metodzie rk45 nie zrobiliśmy porównania, gdyż ta metoda inteligentnie dobiera krok, aby mieć pożądaną błąd, więc nie ma czego tutaj porównywać. Kiedy mamy do czynienia tak jak w naszym przypadku z różniczkami drugiego stopnia, musimy odpowiedni podzielić równanie na dwa osobne. W ten sposób otrzymujemy dwa wyniki - pierwszy po scałkowaniu drugiego stopnia różniczki, a drugi po scałkowaniu otrzymanego pierwszego stopnia różniczki. Przyjrzyjmy się jak to wygląda w kodzie na przykładzie rk2 Heuna.

```
"""IMPLEMENTACJA METODY RK2 HEUNA"""  
  
def RK2(f,x0,t0,tk,h):  
    t=np.arange(t0,tk,h)  
    N=len(t)  
    if hasattr(x0, "__len__"): x=np.zeros((N,len(x0)))  
    else: x=np.zeros(N)  
    x[0]=np.array(x0)  
    i=1  
    while (i<N):  
        k1=h*f(x[i-1], t[i-1])  
        k2=h*f(x[i-1]+k1, t[i-1]+h)  
        x[i]=np.array(x[i-1]+(k1+k2)/2)  
        i+=1  
    return t,x
```

Rysunek 1: Implementacja metody rk2

```

"""RÓWNANIE RUCHU ORAZ ROZBICIE GO NA CZĘŚCI"""
def f(y,t):
    k = 3.94
    p = 2
    m = 0.1

    x = y[0]
    v = y[1]
    dxdt = v
    dvdt = -k*x**(p-1)/m + 50*math.sin(6.28*x)
    return np.array([dxdt, dvdt])

```

Rysunek 2: Implementacja równania ruchu oscylatora harmonicznego

```

"""ROZWIĄZANIE METODĄ ODEINT DLA PORÓWNANIA"""
trange = 5
h = 0.01

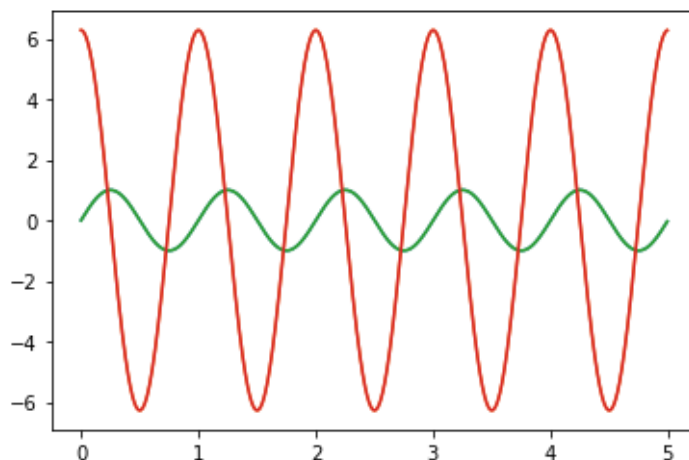
ts = arange(0, trange, h)
x0 = 0.0
v0 = 6.28
y0 = array([x0, v0])

ys = odeint(f, y0, ts)
xs=ys[:,0]
vs=ys[:,1]

```

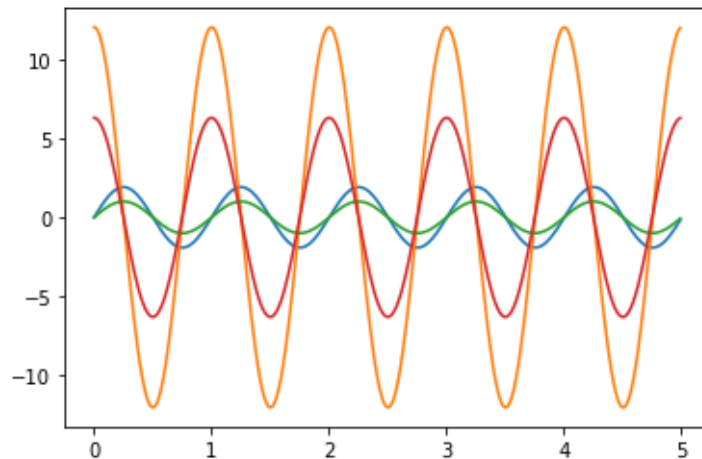
Rysunek 3: Implementacja metody odeint

W następnej kolejności mogliśmy narysować wykres naszych rozwiązań. Oczywiście mowa tutaj o położeniu i prędkości. Na jednym wykresie umieszczone są rozwiązania naszą metodą oraz metodą odeint. Odpowiadają temu pary kolorów pomarańczowy i niebieskie oraz zielony i czerwony. Na wykresie widzimy jedynie zielony i czerwony co świadczy o tym, że wykresy niemalże idealnie się pokrywają. Błąd jest bardzo niskiego rzędu, stąd uzyskujemy właśnie taki efekt.



Rysunek 4: Zależność położenia i prędkości od czasu obliczona obiema metodami

W kolejnych elementach zadania oddaliśmy się rozważaniom co stanie się z wykresem gdy będziemy zmieniać różne parametry. Doszliśmy do kilku ciekawych konkluzji, takich jak rozbieżność rozwiązania numerycznego z naszym przy dużym przedziale czasowym i zbyt małym kroku. Jednak tu w raporcie pozwolimy sobie ominąć te szczegóły związane z dokładnością i zmianami wyglądu wykresu. Kłucznym jednak jest do pokazania, że nasz oscylator jest izochroniczny, czyli że okres jego drgań nie zmienia się wraz z amplitudą. Pokazaliśmy to poprzez zmodyfikowania rozwiązania jedną metodą względem rozwiązania drugą metodą. W efekcie uzyskaliśmy porządany efekt, czyli wykazanie izochroniczności oscylatora. Widać to doskonale na poniższym wykresie.



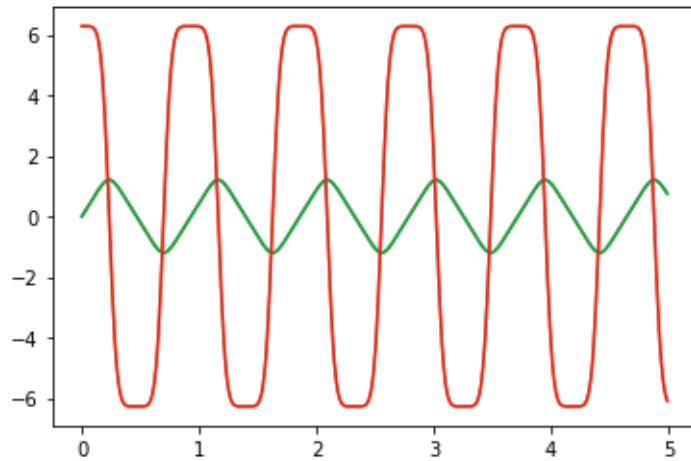
Rysunek 5: Izochroniczność oscylatora harmonicznego

Ostatnim etapem ćwiczenia było sprawdzenie naszych metod na przykładzie dwóch równań:

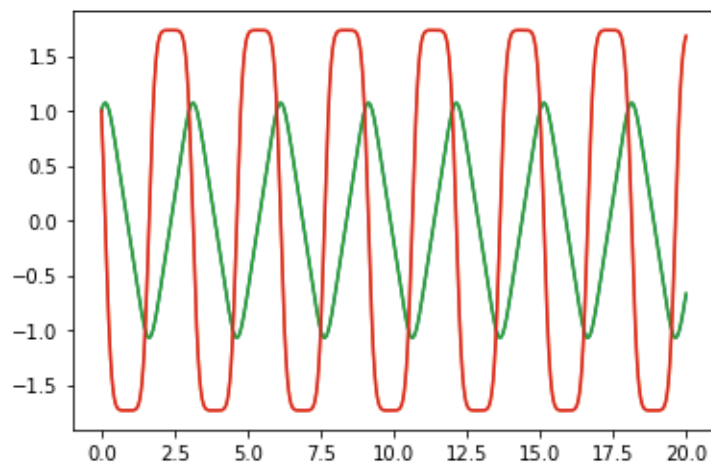
$$2yy'' + y^2 - y'^2 = 0, \quad (1)$$

$$y'' + 6y^5 = 0. \quad (2)$$

Głównym zamysłem było sprawdzenie dokładności naszych metod dla tych równań. Dodatkowo ciekawym faktem jest to, że rozwiązanie równania (2) oraz rozwiązanie równania ruchu oscylatora harmonicznego dla $p = 6$ wykazują ogromne podobieństwo do siebie. Można zaobserwować to na poniższych dwóch grafikach.



Rysunek 6: Rozwiązanie oscylatora harmonicznego dla $p = 6$



Rysunek 7: Rozwiązanie równania (2)

Ostatecznie kończąc nasze rozważania, sporządziliśmy tabelkę z porównaniem wszystkich metod, ukazując jaki błąd został popełniony, w jakim czasie wykonane zostały obliczenia i jaki przyjęliśmy krok. Badaliśmy również co się dzieje z błędem i czasem gdy zmieniamy parametry. Dla metody rk45 krok nie jest wartością stałą, gdyż jest on odpowiednio dobierany przez program w zależności jakiego błędu oczekujemy. Wszystko to przedstawia poniższa tabela.

Przedział [0,5]							
		Specjalny warunek	błąd x	błąd v	czas [s]	h	
oscylator hamroniczny	rk2 Heuna	$p = 2$	6,52E-03	4,14E-02	0,007	0,01	
		$p = 6$	5,90E-02	3,42E-01	0,012		
		dodatkowa siła, $p = 2$	4,28E-03	2,62E-02	0,012		
		dodatkowa siła, $p = 6$	5,23E-02	3,09E-01	0,014		
	rk2 Mid Point	$p = 2$	6,53E-03	4,14E-02	0,0065	0,01	
		$p = 6$	6,55E-03	3,80E-01	0,013		
		dodatkowa siła, $p = 2$	4,68E-03	2,84E-02	0,009		
		dodatkowa siła, $p = 6$	6,20E-02	3,68E-01	0,011		
	rk4	$p = 2$	1,20E-06	7,67E-06	0,02	0,01	
		$p = 6$	9,29E-05	5,44E-04	0,024		
		dodatkowa siła, $p = 2$	2,19E-06	1,33E-05	0,019		
		dodatkowa siła, $p = 6$	7,60E-05	4,56E-04	0,036		
				błąd	czas [s]	h średnie	
	rk45	$p = 2$	6,58E-07		0,047	0,01025	
		$p = 6$	6,87E-07		0,1	0,00463	
		dodatkowa siła, $p = 2$	3,19E-06		0,072	0,00858	
dodatkowa siła, $p = 6$		1,53E-06		0,16	0,00463		
Przedział [0,20]							
			błąd y	błąd y'	czas [s]	h	
równanie (1)	rk4	---	4,51E-07	4,91E-07	0,08	0,01	
	rk45	---	5,60E-07		0,007	0,3127	
	rk2 Heuna	---	3,18E-06	5,22E-06	0,054	0,01	
	rk2 MidPoint	---	3,48E-06	6,74E-06	0,052	0,01	
równanie (2)	rk4	---	2,00E-06	3,52E-06	0,083	0,01	
	rk45	---	1,84E-06		0,059	0,02815	
	rk2 Heuna	---	6,03E-03	1,04E-02	0,05	0,01	
	rk2 MidPoint	---	3,72E-03	6,45E-03	0,07	0,01	

Rysunek 8: Tabelka porównawcza metod

4. Podsumowanie

Realizując projekt zapoznaliśmy się ze sposobem działania metod Rungego-Kutty. Dochodzimy do wniosku, że są one potężnym narzędziem matematycznym. Przy odrobinie mocy obliczeniowej i odpowiednim użyciu równań metody możemy uzyskać bardzo dokładne całkowanie w szybkim tempie. Jedynym minusem metod Rungego-Kutty jest brak możliwości wykonania obliczeń na kartce papieru. Ciężko sobie wyobrazić sytuację, w której musielibyśmy dla kilkudziesięciu tysięcy kroków wykonywać po kilka linijek żmudnych obliczeń. Jednak taki minus nie znaczy wiele w porównaniu z możliwościami jakie metody te oferują. W trakcie projektu mogliśmy się przekonać jednocześnie o błędach jakie taka metoda może popełnić. Czasem całkowanie zostaje wykonane poprawnie, jednak rozwiązanie może różnić się od analitycznego w niektórych przypadkach.

5. Bibliografia

W raporcie nie posłużyliśmy się żadnymi źródłami zewnętrznymi. Cały tekst został napisany samodzielnie, a obrazki wzięte są z naszego projektu.