Z metodą Rungego–Kutty przez przestrzeń fazową

Ryszard KUTNER

Spośród wielu zagadnień fizyki, w których badaniu ważną rolę odgrywają metody numeryczne, na czoło wysuwa się problem ewolucji czasowej wybranych układów fizycznych. Mogą to być np. punkty materialne lub ich układy, płyny, ale także skomplikowane układy kwantowe. W artykule tym ograniczymy się do najprostszego przypadku – punktów materialnych. Wówczas ewolucja, czyli w tym przypadku po prostu ruch, jest rozwiązaniem układu N (tylu, ile jest cząstek) równań, z których każde jest postaci

$$m_i \frac{\mathrm{d}^2}{\mathrm{d}t^2}\vec{r_i} = \vec{F_i}\left(\vec{r_1},\ldots,\vec{r_N},\frac{\mathrm{d}\vec{r_1}}{\mathrm{d}t},\ldots,\frac{\mathrm{d}\vec{r_N}}{\mathrm{d}t},t\right), \tag{1}$$

przy czym siły $\vec{F_i}$ traktujemy jako dane funkcje położeń, prędkości i czasu.
Do rozwiązania metodą numeryczną postać równań (1) nie jest zbyt wygodna – znacznie lepiej mieć do czynienia z równaniami rzędu pierwszego. Można to osiągnąć bardzo prosto, wprowadzając jako dodatkowe funkcje nieznane pędy każdej z cząstek

$$\vec{p_i} = m_i \frac{\mathrm{d}\vec{r_i}}{\mathrm{d}t}, \tag{2}$$

za cenę podwojenia liczby równań. Rozpisując jeszcze położenia i pędy na składowe możemy zapisać tak otrzymany układ $6N$ równań w zwartej postaci

$$\frac{\mathrm{d}}{\mathrm{d}t}\underline{u}(t) = \underline{f}\left[\underline{u}(t),t\right], \tag{3}$$

gdzie użycie symboli podkreślonych oznacza, że chodzi nam o wektory o $k = 6N$ składowych (np. $\underline{u} = (r_{1x}, p_{1x}, r_{1y}, p_{1y}, \ldots, r_{Nx}, p_{Nx})$).

Okazuje się, że opis układów fizycznych, w którym położenia i pędy występują na równych prawach, jako niezależne wielkości, jest użyteczny nie tylko przy numerycznym rozwiązywaniu równań ruchu. Co więcej, sformułowanie takie stoi u podstaw całych wielkich działów fizyki, takich jak fizyka statystyczna i mechanika kwantowa. Także w ramach mechaniki klasycznej formalny brak rozróżnienia między położeniem a pędem pozwala na wyprowadzenie zupełnie nowej metody rozwiązywania zagadnień dynamiki (jest to tzw. równanie Hamiltona–Jacobiego). Szczególną rolę w tych wszystkich rozważaniach odgrywa pojęcie przestrzeni fazowej. Jest to przestrzeń, której punktami są wszystkie możliwe stany układu sparametryzowane wartościami położeń i pędów. Jeżeli w pewnej chwili t_0 układ mechaniczny scharakteryzowany był pewnymi wartościami położeń i pędów $(\underline{x}_0, \underline{p}_0)$, to obrazem jego ruchu w przestrzeni fazowej jest krzywa przechodząca przez punkt o współrzędnych $(\underline{x}_0, \underline{p}_0)$. Krzywą taką nazywamy trajektorią fazową. Okazuje się, że **trajektorie fazowe scharakteryzowane różnymi wartościami początkowymi $(\underline{x}_0, \underline{p}_0)$ nie przecinają się.** Jest to podstawowa zaleta opisu fazowego w porównaniu ze zwykłym, w którym śledzimy tylko zmiany położeń (nawiasem mówiąc, przestrzeń położeń nosi fachową nazwę przestrzeni konfiguracyjnej) – te same położenia mogą zostać osiągnięte tutaj przy różnych warunkach początkowych. Rozłączność trajektorii fazowych pozwala na ciekawą interpretację ruchu w przestrzeni fazowej jako przepływu pewnej hipotetycznej cieczy. Istotnie, każdy punkt przestrzeni fazowej możemy związać z pewnymi warunkami początkowymi dla naszego układu. Wraz z upływem czasu położenia punktów będą się zmieniać zgodnie z równaniami ruchu, jednak punkty – podobnie jak cząsteczki cieczy – zachowają zawsze swoją identyczność. A zatem jednoczesnemu ruchowi wszystkich możliwych kopii danego układu mechanicznego odpowiada przepływ „cieczy fazowej". Taka ciecz, choć wielowymiarowa, zachowuje się analogicznie jak zwykła woda. Na przykład – jest ona nieściśliwa.

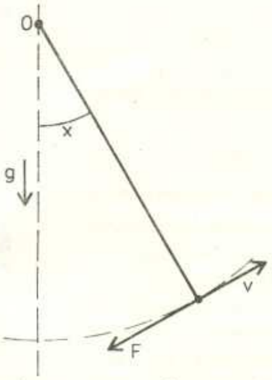
Wróćmy jednak do problemu numerycznego rozwiązania równania ewolucji (3) – nagrodą będzie możliwość bliższego zaznajomienia się z cieczą fazową. Jak zwykle w takich przypadkach, wprowadzimy szereg równo odległych węzłów czasowych i będziemy szukać algorytmu, pozwalającego wyznaczyć rozwiązanie $\underline{u}(t_{n+1})$ w węźle t_{n+1} korzystając ze znajomości rozwiązania $\underline{u}(t_n)$ w węźle poprzednim t_n.

```
  10 REM Program symuluje ruch
         punktu w przestrzeni
                      fazowej
  20 REM METODA RUNGEGO-KUTTY
  30 REM
  40 DIM x(1001): REM Położenie
             w węźle głównym
  50 DIM v(1001): REM Prędkość
             w węźle głównym
  60 REM t - Czas
  70 REM dt - Elementarny
            przedział czasu
  80 REM INT (tk/dt) - Całkowita
     liczba elementarnych kroków
                     czasowych
  90 REM
 100 LET m=1:
     LET g=9.81:
     LET l=9:
     LET ro=0
 110 REM
 120 REM F - Siła
 130 REM
 140 DEF FN F(x,v,t)=-m*g*SIN x
                      /l-ro*v/l
 150 REM
 160 LET dt=0.1
 170 LET tk=350*dt
 180 REM x0 - Położenie począt-
                          kowe
 190 REM v0 - Prędkość
                   początkowa
 200 LET licz=0
 210 INPUT "x0=";x0
 220 IF ABS x0>2↑31 THEN
                   GO TO 210
 230 INPUT "v0=";v0
 240 IF ABS v0>3e8 THEN GO TO
                         230
 250 LET x=x0: LET v=v0
 260 LET licz=licz+1
 270 PRINT AT 0,25;"OBLICZA"
 280 REM
 290 FOR n=1 TO INT (tk/dt)+1
 300     LET t=(n-1)*dt
 310     LET x(n)=x
 320     LET v(n)=v
 330     LET a=FN F(x,v,t)/m
 340 REM Równania pomocnicze
 350     LET ts=t+dt/2
 360     LET tn=t+dt
 370     LET k1x=v
 380     LET k1v=a
 390     LET k2x=v+dt*k1v/2
 400     LET x2=x+dt*k1x/2
 410     LET v2=v+dt*k1v/2
 420     LET k2v=FN F(x2,v2,ts)/
                             m
 430     LET k3x=v+dt*k2v/2
 440     LET x3=x+dt*k2x/2
 450     LET v3=v+dt*k2v/2
 460     LET k3v=FN F(x3,v3,ts)/
                             m
 470     LET k4x=v+dt*k3v
 480     LET x4=x+dt*k3x
 490     LET v4=v+dt*k3v
 500     LET k4v=FN F(x4,v4,tn)/
                             m
 510 REM Równania główne
 520     LET x=x+dt*(k1x+2*k2x+2
                  *k3x+k4x)/6
 530     LET v=v+dt*(k1v+2*k2v+2
                  *k3v+k4v)/6
 540 NEXT n
 550 REM
 560 IF licz>1 THEN GO TO 810
 570 REM
 580 REM WYKRES v(x)
 590 REM
 600 LET s=64
 610 INPUT "sx=";sx
 620 INPUT "sv=";sv
 630 PRINT AT 0,0;" v↑[*";INT
          (100*sv+.5)/100;"]"
 640 PRINT " 1": PRINT : PRINT :
     PRINT
 650 PRINT : PRINT : PRINT :
     PRINT : PRINT " 0"
 660 PRINT : PRINT : PRINT :
     PRINT : PRINT
 670 PRINT : PRINT : PRINT "-1"
 680 PRINT : PRINT TAB 22;">"
 690 PRINT TAB 3;"-1";TAB 12;"0"
                ;TAB 20;"1";
 700 PRINT TAB 22;"x[*";INT (100
             *sx+.5)/100;"]"
 710 PLOT 0,19: DRAW 180,0
 720 PLOT 19,0: DRAW 0,170
 730 REM
 740 FOR i=4 TO 20 STEP 2
       PLOT 8*i+3,19: DRAW 0,3
 750 NEXT i
 770 REM
 780 FOR j=4 TO 20 STEP 2
 790   PLOT 19,8*j+3: DRAW 3,0
 800 NEXT j
 810 PRINT AT 0,25;"       "
 820 FOR n=0 TO INT (tk/dt)
 830     LET x=x(n+1)/sx*s+99
 840     IF x>163 OR x<35 THEN
                   GO TO 210
 850     LET v=v(n+1)/sv*s+99
 860     IF v>163 OR v<35 THEN
                   GO TO 210
 870     PLOT INT x,INT v
 880 NEXT n
 890 GO TO 210
```

Na pierwszy rzut oka jest to dość łatwe. Ponieważ $\frac{d}{dt}\underline{u}(t_n) \simeq \frac{\Delta\underline{u}(t_{n+1})}{\Delta t} = \frac{\underline{u}(t_{n+1})-\underline{u}(t_n)}{\Delta t}$, więc

$$(4) \qquad \underline{u}(t_{n+1}) \simeq \underline{f}[\underline{u}(t_n), t_n]\Delta t + \underline{u}(t_n)\,.$$

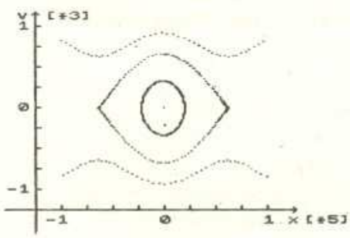
Niestety, takie proste rozwiązanie prowadzi do nieprzewidzianych kłopotów. Przy używaniu elementarnych algorytmów, takich jak opisany powyżej, obserwować można niefizyczny efekt samoczynnego numerycznego „studzenia się" lub „grzania", co prowadzi do niestabilnego zachowania się układu przekreślając wiarygodność stosowanej metody. Efekt ten polega na tym, że całkowita energia układu maleje lub rośnie bez jakichkolwiek fizycznych przyczyn – sama z siebie. Powodem takiego niefizycznego zachowania się są kumulujące się błędy procedury numerycznej (szerzej zostało to opisane w książce D. Pottera *Metody obliczeniowe fizyki – fizyka komputerowa*, PWN, Warszawa 1982). Zwykle, aby temu zaradzić, w pierwszym odruchu zmniejszamy krok czasowy. Prowadzi to jednak do spowolnienia metody, a wspomniane efekty nadal się pojawiają, tyle że nieco później. Nie tędy więc droga. Należy raczej użyć bardziej wyrafinowanego algorytmu obarczonego mniejszym błędem i stabilnego ze względu na ten błąd. Algorytmem spełniającym te wymagania jest tzw. metoda Rungego–Kutty. Jej zwykłe wyprowadzenie jest dość zawiłe. Można ją jednak również otrzymać różnicując równanie różniczkowe (3) na trzy różne sposoby, a następnie – traktując każdy z nich jako równoprawny – uśrednić względem tych sposobów (patrz tekst w ramce na str. 8). Rezultatem jest w pełni profesjonalny algorytm o bardzo małym błędzie (rzędu $(\Delta t)^5$), stosowany z powodzeniem do rozwiązywania skomplikowanych zagadnień ewolucji. My jednak nie będziemy tak ambitni i zadowolimy się analizą przypadku najprostszego – jednowymiarowego ruchu pojedynczego punktu materialnego. Naszym wzorcowym układem będzie wahadło matematyczne (rys. 1). Aby jeszcze bardziej uprościć sobie życie, przyjmiemy, że masa punktu jest jednostkowa. Wówczas $p = v$ i mamy $\underline{u}(t) = [x(t), v(t)]$, $\underline{f}[\underline{u}(t); t] = \{v(t), F(x(t), v(t); t)\}$. Załączony program komputerowy znajduje trajektorie fazowe dla tego prostego układu. Został on napisany w BASICu na mikrokomputer ZX Spektrum 48 K (lub jemu pokrewne, jak Timex czy Elwro 800 Junior), a ze względu na jego prostotę można go także bez trudu przepisać na inne, popularne w kraju mikrokomputery. Po uruchomieniu program zapytuje o początkowe wartości położenia i prędkości oraz o czynniki skalujące poziomą oś x–ów i pionową oś v (zostają one następnie wydrukowane na wykresie – patrz rysunek 2 i 3). Po obliczeniu trajektorii fazowej pytanie o warunki początkowe zostaje ponowione (czynników skali zmieniać już wtedy nie można).



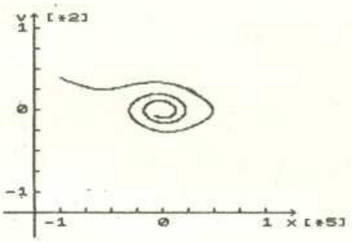
$\frac{dv}{dt} = -\frac{g}{l}\sin x - \frac{\varrho}{ml} v = F/ml$

$\alpha \equiv \varrho$ — współczynnik oporu ośrodka

Rys. 1



Rys. 2



Rys. 3

Rysunki 2 i 3 ilustrują wynik działania programu dla zagadnienia wahadła matematycznego, odpowiednio przy braku tłumienia i z uwzględnieniem siły tłumiącej – patrz rysunek 1. Otrzymany wzór przypomina nieco dywan, a nieco ścieżkę wirów powstającą za przeszkodą w płynącej wodzie (szczególnie rysunek 2). Górny „szlaczek" na rysunku 2, podobnie jak i dolny, to tzw. przypadek rotacyjny (zachodzący tylko wtedy, gdy $k = \frac{|v_0|}{\omega} > 1$, gdzie częstość $\omega = \sqrt{g/l}$), gdy wahadło wiruje wokół swojego punktu zawieszenia 0. Spoczynkowi wahadła odpowiada po prostu kropka zaznaczona na wykresie w punkcie (0,0). Następnie elipsa odpowiada przypadkowi oscylacyjnemu ($k < 1$). Wreszcie przypadek graniczny ($k = 1$) niezwykle trudny do wysymulowania, odpowiadający zatrzymaniu się wahadła w pozycji pionowej ($x = \pi$). Trudność numeryczna polega na tym, iż wahadło w swoim ruchu ma się zatrzymać dokładnie w zenicie i w takiej pozycji pozostać. Jednakże nawet najmniejszy błąd numeryczny (związany np. ze skończoną długością przedziału czasowego dt) wytrąca je prędzej czy później z tego stanu. (Niezłym przybliżeniem jest przyjęcie przedziału czasowego $dt = 0,1$.)

Proponujemy Czytelnikowi odtworzenie tych rysunków. Aby to umożliwić, jak również, by otworzyć drogę do własnych eksperymentów, potrzeba jeszcze kilku słów o strukturze programu. W linii 140 zdefiniowano siłę F, a w linii 100 przyjęto parametry ją charakteryzujące. Zasadniczy algorytm zawarty jest w (wydzielonej) pętli rozpoczynającej się w linii 290, a kończącej się w linii 540. Zidentyfikowanie metody Rungego–Kutty nie powinno tu stanowić problemu. Inne podstawowe parametry, charakteryzujące samą symulację, to elementarny przedział czasu dt ($\equiv \Delta t$) ustalony w linii 160 oraz całkowity czas symulacji t_k (linia 170).

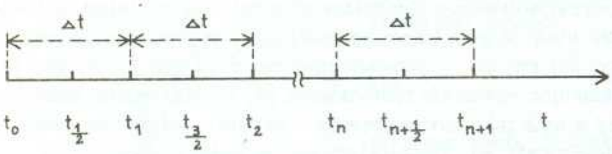
A zatem wszystko jest już gotowe do wyprawy w przestrzeń fazową. Szczęśliwej podróży!

Opis metody Rungego–Kutty

Chcemy rozwiązać równanie różniczkowe zwyczajne pierwszego rzędu

$$\frac{\mathrm{d}}{\mathrm{d}t}\underline{u}(t) = \underline{f}[\underline{u}(t), t]\,.$$

W pierwszym kroku przybliżymy pochodną względem czasu $\frac{\mathrm{d}}{\mathrm{d}t}\underline{u}(t)$ przez iloraz $\frac{\Delta \underline{u}(t)}{\Delta t} \approx \frac{\underline{u}(t_{n+1})-\underline{u}(t_n)}{\Delta t}$, gdzie indeks $n = 0, 1, 2, \ldots$, numeruje kolejne (w tym przypadku) równo odległe węzły czasowe, tak jak to pokazuje poniższy rysunek.



Będziemy poszukiwać rozwiązania właśnie w tych (głównych) węzłach czasowych. Kłopot jednak w tym, że iloraz $\frac{\underline{u}(t_{n+1})-\underline{u}(t_n)}{\Delta t}$ wyznacza pewną funkcję (prędkość) w połowie przedziału czasowego, czyli w węźle pomocniczym $t_{n+(1/2)}$. A zatem możemy wypisać następujące przybliżone równości:

$$\frac{\underline{u}(t_{n+1}) - \underline{u}(t_n)}{\Delta t} \approx \frac{1}{2}\{\underline{f}[\underline{u}(t_n), t_n] + \underline{f}[\underline{u}(t_{n+1}), t_{n+1}]\}\,,$$

$$\frac{\underline{u}(t_{n+1}) - \underline{u}(t_n)}{\Delta t} \approx \underline{f}[\underline{u}(t_{n+(1/2)}), t_{n+(1/2)}]\,,$$

$$\frac{\underline{u}(t_{n+1}) - \underline{u}(t_n)}{\Delta t} \approx \underline{f}\left[\frac{\underline{u}(t_n) + \underline{u}(t_{n+1})}{2}, t_{n+(1/2)}\right]\,.$$

Stanowią one równoprawne różnicowe wersje naszego wyjściowego równania różniczkowego. I nie ma nic zaskakującego w tym, że wiele różnych obrazów różnicowych przybliża jedno i to samo równanie różniczkowe. Do tego trzeba się po prostu przyzwyczaić. Przypomnijmy, **że celem naszym jest zbudowanie algorytmu, który wyznacza rozwiązanie $\underline{u}(t_{n+1})$ w następnym węźle czasowym t_{n+1} korzystając ze znajomości rozwiązania $\underline{u}(t_n)$ w węźle poprzednim** t_n. Narzuca się jednak pytanie: w jaki sposób poradzić sobie z prawymi stronami powyższych równań, skoro występuje tam zarówno poszukiwane rozwiązanie $\underline{u}(t_{n+1})$ w węźle następnym, jak też nieznane rozwiązanie $\underline{u}(t_{n+(1/2)})$ w pomocniczym węźle czasowym? Aby rozwiązać ten problem, wprowadźmy następujące pomocnicze wielkości:

$$\underline{k}_1 = \underline{f}[\underline{u}(t_n), t_n]\,,$$

$$\underline{k}_2 = \underline{f}[\underline{u}(t_{n+(1/2)}), t_{n+(1/2)}]\,,$$

$$\underline{k}_3 = \underline{f}\left[\frac{\underline{u}(t_n) + \underline{u}(t_{n+1})}{2}, t_{n+(1/2)}\right]\,,$$

$$\underline{k}_4 = \underline{f}[\underline{u}(t_{n+1}), t_{n+1}]\,,$$

z których, jak widać, jedynie $\underline{k}_1$ możemy od razu wyznaczyć. Pozostałe będziemy teraz mozolnie obliczać. Po pierwsze, możemy napisać w przybliżeniu:

$$\underline{u}(t_{n+(1/2)}) = \underline{u}\left(t_n + \frac{\Delta t}{2}\right) \approx \underline{u}(t_n) + \frac{\Delta t}{2}\underline{f}[\underline{u}(t_n), t_n]\,,$$

a co za tym idzie,

$$\underline{k}_2 = \underline{f}\left[\underline{u}(t_n) + \frac{\Delta t}{2}\underline{k}_1, t_n + \frac{\Delta t}{2}\right]\,.$$

Tym samym wielkość tę potrafimy już obliczyć, gdyż wiemy, jak funkcja $\underline{f}$ zależy od swoich argumentów. W dalszym ciągu zauważmy, że

$$\underline{k}_3 = \underline{f}\left[\underline{u}(t_n) + \frac{1}{2}\Delta t\underline{k}_2, t_n + \frac{\Delta t}{2}\right]$$

(kto nie wierzy, niech podstawi tutaj zamiast $\underline{k}_2$ po prostu iloraz $\frac{\underline{u}(t_{n+1})-\underline{u}(t_n)}{\Delta t}$). I wreszcie

$$\underline{k}_4 = \underline{f}[\underline{u}(t_n) + \Delta t\underline{k}_3, t_n + \Delta t]$$

(co widać po podstawieniu , podobnie jak i poprzednio w miejsce $\underline{k}_3$ ilorazu $\frac{\underline{u}(t_{n+1})-\underline{u}(t_n)}{\Delta t}$). Podsumujmy pracę, jaką wykonaliśmy dotychczas i zastanówmy się, które spośród trzech wyjściowych równań będziemy w dalszym ciągu eksploatować? Odpowiedź jest prosta: wszystkie trzy, gdyż są jednakowo dobre. Jak to praktycznie robić? Ależ to proste – dodajmy je wszystkie i podzielmy przez ich liczbę, czyli, jak to się mówi, uśrednijmy po wszystkich możliwych sposobach (sytuacjach). W wyniku takiego postępowania otrzymujemy gotowy algorytm postaci:

$$\underline{u}(t_{n+1}) = \underline{u}(t_n) + \frac{1}{6}\Delta t(\underline{k}_1 + 2\underline{k}_2 + 2\underline{k}_3 + \underline{k}_4), \quad n = 0, 1, \ldots,$$

gdzie $\underline{k}_j$, $j = 1, \ldots, 4$, zostały już obliczone powyżej. Proszę zauważyć, że za pomocą tego algorytmu możemy znaleźć poszukiwane rozwiązanie w następnym węźle czasowym korzystając tylko ze znajomości rozwiązania w węźle poprzedzającym. Startując więc ze znanego warunku początkowego $\underline{u}(t_0)$ możemy teraz kolejno, krok po kroku, wyznaczyć rozwiązanie w chwili t_1, $\underline{u}(t_1)$, a stąd w chwili t_2 mamy $\underline{u}(t_2)$, po czym $\underline{u}(t_3)$ w chwili t_3, itd. **Osiągnęliśmy zatem postawiony cel.**

Dla zagadnienia jednowymiarowego ruchu punktu materialnego o masie m pod wpływem działania siły $F(x, v; t)$ należy w równaniu (3) przyjąć $\underline{u}(t) = [x(t), v(t)]$ oraz $\underline{f}[\underline{u}(t); t] = \left\{v(t), \frac{F[x(t),v(t);t]}{m}\right\}$. Pomocnicze funkcje k_j, $j = 1, \ldots, 4$, konieczne przy zastosowaniu metody Rungego–Kutty, przyjmą postać:

$\underline{k}_1 = (k_1^x, k_1^v)$, gdzie

$$k_1^x = v(t_n),$$

$$k_1^v = \frac{F[x(t_n), v(t_n); t_n]}{m}\,,$$

$\underline{k}_2 = (k_2^x, k_2^v)$, gdzie

$$k_2^x = v(t_n) + \frac{1}{2}\Delta t k_1^v,$$

$$k_2^v = \frac{F\left[x(t_n) + \frac{1}{2}\Delta t k_1^x, v(t_n) + \frac{1}{2}\Delta t k_1^v, t_n + \frac{\Delta t}{2}\right]}{m}$$

$\underline{k}_3 = (k_3^x, k_3^v)$, gdzie

$$k_3^x = v(t_n) + \frac{1}{2}\Delta t k_2^v,$$

$$k_3^v = \frac{F\left[x(t_n) + \frac{1}{2}\Delta t k_2^x, v(t_n) + \frac{1}{2}\Delta t k_2^v, t_n + \frac{\Delta t}{2}\right]}{m}$$

$\underline{k}_4 = (k_4^x, k_4^v)$, gdzie

$$k_4^x = v(t_n) + \Delta t k_3^v,$$

$$k_4^v = \frac{F\left[x(t_n) + \Delta t k_3^x, v(t_n) + \Delta t k_3^v, t_n + \Delta t\right]}{m}\,.$$