

Opis metod numerycznych dla równań różniczkowych

Materiały do laboratorium z przedmiotu „Wybrane zagadnienia z matematyki”

*Autor: Paweł Żak
(pawelzak@agh.edu.pl)*

Data ostatniej modyfikacji: 10 stycznia 2011

Spis treści

1	Wstęp	2
2	MAXIMA – podstawowe funkcje programu	3
2.1	Maxima jako kalkulator	3
2.2	Wyrażenia algebraiczne i równania.....	6
2.2.1	Wprowadzanie wyrażen algebraicznych	6
2.2.2	Przypisywanie wartości zmiennym.....	6
2.2.3	Definiowanie i rozwiązywanie równań.....	7
3	Definicja równania różniczkowego zwyczajnego.....	8
3.1	Definicja.....	8
3.1.1	Warunki początkowe i brzegowe	9
4	Równania różniczkowe zwyczajne rzędu pierwszego i drugiego.....	9
4.1	Postaci równań.....	9
4.1.1	Równania w postaciach rozwikłanych.....	9

4.1.2	Zagadnienia początkowe (<i>zagadnienie Cauchy'ego</i>).....	10
4.1.3	Zagadnienie brzegowe dla równania różniczkowego rzędu drugiego	10
4.2	Przykłady równań różniczkowych.....	10
4.2.1	Wahadło matematyczne	10
4.2.2	Poszukiwanie rozwiązania dokładnego problemu drgań wahadła matematycznego... ..	12
4.2.3	Prezentacja uzyskanych wyników	14
5	Schematy różnicowe dla równań różniczkowych zwyczajnych	17
5.1	Obserwacja wrażliwości równania różniczkowego na wartości początkowe oraz na parametry modelu.....	17
5.1.1	Pole kierunków związane z równaniem różniczkowym	17
5.1.2	Wrażliwość rozwiązania na wybór warunku początkowego	18
5.1.3	Wrażliwość rozwiązania na parametry modelu	19
5.1.4	Ćwiczenia.....	20
5.2	Schemat Eulera.....	20
5.2.1	Wyprowadzenie schematu przy wykorzystaniu interpretacji geometrycznej	20
5.2.2	Przykład obliczeń schematem Eulera	21
6	Dodatek A	22
6.1	Definicje.....	22
6.1.1	Radian.....	22
6.1.2	Wahadło matematyczne	23
6.1.3	Wzór Taylora i rozwinięcie funkcji w szereg potęgowy.....	23

1 Wstęp

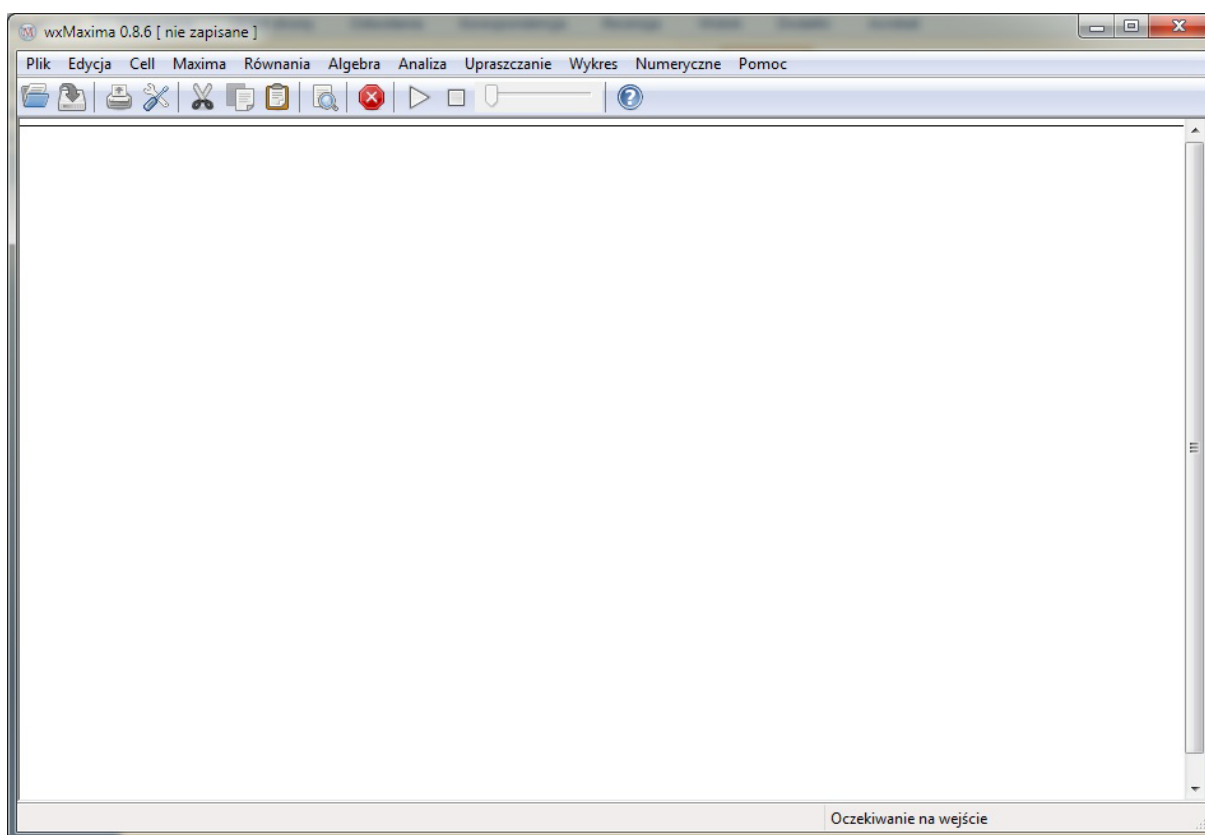
Poniższe materiały zostały przygotowane z myślą o uczestnikach laboratoriów z przedmiotu „Wybrane zagadnienia z matematyki”. W zbiorze tym postaram się zgromadzić opis zagadnień poruszanych podczas ćwiczeń, by umożliwić słuchaczom łatwiejsze przyswojenie wiedzy.

Podczas zajęć używamy programu MAXIMA. Jest to program opensource. Aby uzyskać dostęp do kopii tego programu lub oraz do materiałów szkoleniowych proponuję odwiedzić stronę: <http://maxima.sourceforge.net/>

2 MAXIMA – podstawowe funkcje programu

Maxima jest programem służącym do wykonywania obliczeń symbolicznych. Można go używać do rozwiązywania równań, przekształcania wzorów, jako środowisko, w którym programujemy nieskomplikowane algorytmy, może mieć jeszcze wiele zastosowań między innymi może być stosowana jako zwykły kalkulator.

Po zainstalowaniu w menu start pojawia się dodatkowy folder zawierający skróty. Znajdują się tam, między innymi, skróty umożliwiające dostęp do jednej z dwóch wersji programu wxMaxima oraz XMaxima. Funkcjonalność obu wspomnianych wersji jest taka sama jednak interface programu wxMaxima wydaje się być bardziej przyjazny. W tym opisie wszelkie zrzuty ekranu będą właśnie z tej wersji.



Rysunek 2-1. Obszar roboczy programu wxMaxima. Białe okno, to obszar w którym prowadzimy dialog z programem. Menu występujące w górnej części okna pozwala na podstawowe operacje dyskowe oraz na umożliwia wykonanie poleceń w uproszczony sposób.

2.1 Maxima jako kalkulator

Program wxMaxima może być wykorzystywany do przeprowadzania podstawowych operacji arytmetycznych z wykorzystaniem zmiennych. Praca z programem odbywa się na zasadzie dialogu, w którym po wysłaniu zapytania do jądra systemu na ekranie pojawia się odpowiedź lub komunikat o niemożności zwrócenia odpowiedzi. Przykładem takiego zapytania może być próba wyznaczenia sumy dwóch liczb:

```
(%i1) 2.35+6;  
(%o1) 8.35
```

Pytania wprowadzone przez użytkownika oznaczane są symbolem `%i_numer`, odpowiedzi `%o_numer`. Wysłanie zapytania do jądra systemu obliczeń symbolicznych odbywa się po naciśnięciu kombinacji klawiszy Shift + Enter. Sam klawisz Enter przesuwa nas do nowej linii. W `wxMaxima` w celu oznaczenia części dziesiętnej liczby używamy kropki. Jest to związane z tym, że program został stworzony w Stanach Zjednoczonych, gdzie taka nomenklatura jest obowiązująca. Ewentualne przemieszczenia kodu do nowej linii oraz nadmiarowe odstępki nie mają znaczenia. Instrukcja kończy się wraz z napotkaniem przez program średnika lub dojścia do końca wprowadzanego przez użytkownika tekstu. Wówczas średnik na końcu jest dodawany automatycznie, gdyż jest wymagany przez jądro do poprawnego wykonania operacji. Jeżeli chcemy wykonać więcej niż jedną instrukcję po jednorazowym naciśnięciu kombinacji klawiszy Shift+Enter instrukcje rozdzielamy średnikami:

```
(%i1) 2+3.4;
      3^2 - 5.01;
      5^3 - 5**3;
(%o1) 5.4
(%o2) 3.99
(%o3) 0
```

Z poprzedniego przykładu widać, że podnoszenie do potęgi można zrealizować na jeden z dwóch sposobów. Pierwszym jest operacja analogiczna do tych stosowanych na przykład w arkuszu kalkulacyjnym Excel: \wedge (od symbolu strzałki do góry \uparrow – język programowania Algol). Symbol ten uzyskujemy po równoczesnym wciśnięciu klawiszy Shift + 6. Kolejny sposób pochodzi od języków programowania niskiego poziomu jak FORTRAN: `**`. Operator składa się z dwóch symboli mnożenia (popularnie zwanych „gwiazdkami”). Symbol mnożenia uzyskać możemy wciskając jednocześnie klawisz Shift + 8 lub wybierając odpowiedni klawisz na klawiaturze numerycznej.

W programie można również używać funkcji elementarnych, takich jak pierwiastek, logarytm, funkcje trygonometryczne... Do tego konieczna jest znajomość składni – obowiązujących nazw tych funkcji. Wszystkie je można znaleźć w pomocy programu. Tutaj przytoczymy kilka przykładów.

```
(%i1) sin(2)+cos(2);
      tan(1)-cot(3);
      exp(0);
      log(1);
(%o1) sin(2)+cos(2)
(%o2) tan(1)-cot(3)
(%o3) 1
(%o4) 0
```

Jak łatwo się domyśleć, funkcje do których się odwoływaliśmy w tym przykładzie, to : sinus (`sin`), cosinus (`cos`), tangens (`tan`), cotangens (`cot`), funkcja wykładnicza

o podstawie e (`exp`), funkcja logarytmiczna o podstawie e (`log`). W programie nie zostały zdefiniowane logarytmy o innych podstawach, zatem użyteczna może być wiedza o sprowadzaniu logarytmu naturalnego (o podstawie e) do logarytmu o dowolnej podstawie:

$$\log_a(b) = \frac{\ln(b)}{\ln(a)} \quad (2.1)$$

przez `ln` rozumiemy logarytm o podstawie e .

Pierwiastkowanie możemy uzyskać przed podnoszenie do odpowiednich ułamkowych potęg. Pierwiastek kwadratowy można także uzyskać stosując polecenie `sqrt`.

```
(%i1) sqrt(2);
      3**(1/2);
(%o1)  $\sqrt{2}$ 
(%o2)  $\sqrt{3}$ 

(%i3) 5^(1/3);
(%o3)  $5^{1/3}$ 
```

Obliczenia symboliczne są bardzo dobre, gdyż podczas kolejnych operacji komputer nie popełnia błędu związanego z przybliżeniem liczby. Co jednak, gdy wynikiem ma być interpretowalna wartość? Częstokroć zagnieżdżone operacje matematyczne do tego stopnia komplikują wyrażenie, że nie jest możliwym nawet oszacowanie jego wartości. Wówczas można poprosić program o podanie wartości przybliżonej wyniku przeprowadzonych obliczeń.

```
(%i1) sqrt( exp( sin(2)-cos(sqrt(5))/3 ) * log(4)/65.43 );
      
$$\frac{\sin(2) - \frac{\cos(\sqrt{5})}{3}}{2} \sqrt{\log(4)}$$

(%o1) 0.12362649025871 %e

(%i2) %, numer;
(%o2) 0.25419698569668

(%i3) %o1, numer;
(%o3) 0.25419698569668
```

W powyższym przykładzie można zauważyć, że ciężko byłoby oszacować wartość wyrażenia wprowadzonego przez użytkownika (`%i1`), mimo iż wygląda imponująco. Aby uzyskać wartość tego wyrażenia umieszczamy po nim przecinek i wywołujemy polecenie `numer`. Nakazuje ono zwrócenie wartości przybliżonej wyrażenia znajdującego się przed przecinkiem. W wyrażeniu widzimy symbol `%`, zastępuje on ostatnie wysłane przez program wyrażenie. Do wyrażeń wprowadzonych przez użytkownika, jak również zwróconych przez program można się zawsze odnosić podczas dialogu poprzez podanie `%i_numer` lub `%o_numer`. Nie ważne jak dawno temu wyrażenie to byłoby tematem dyskusji.

wxMaxima umożliwia również kopiowanie fragmentów wprowadzonego już kodu i wklejanie go podczas komponowania kolejnych zapytań. Po wklejeniu wyrażenie będzie zgodne z tymi zasadami formułowania zapytań przez użytkownika. Mimo, że podczas podawania odpowiedzi program stara się jak najwięcej wyrażen zapisywać w przyjaznej formie. Zgodnie z tym w jaki sposób zapisujemy wyrażenia matematyczne. Choć występują pewne modyfikacje, jak np. liczba e zapisywana jest przy pomocy %e. Podobny symbol jest używany podczas wprowadzania π , służy do tego:

%pi

Warto pamiętać, że podczas zagęszczania wyrażen zawsze stosowane są nawiasy okrągłe, tak jak podczas wywoływania funkcji elementarnych.

2.2 Wyrażenia algebraiczne i równania

W programie wxMaxima istnieje możliwość deklarowania wyrażen algebraicznych. Można zapisać nawet bardzo skomplikowane wyrażenia – wzory. Wyrażenia te można dowolnie przekształcać, a także można znaleźć ich wartości przy pewnych, ustalonych wartościach parametrów. Istnieje też możliwość definiowania funkcji oraz przypisywania całych wyrażen pod pewnymi nazwami.

2.2.1 Wprowadzanie wyrażen algebraicznych

Wyrażenia algebraiczne wprowadzamy bezpośrednio z klawiatury. Dozwolone są wszystkie litery jak i ich kombinacje.

```
(%i1) a+b*(c-a);
(%o1) b(c-a)+a

(%i2) (a-b)*(a+b)+(a-3)*(%pi+b)^2;
(%o2) (a-3)(b+π)2+(a-b)(b+a)

(%i3) x+y**2-(a*c+2*c)^3;
(%o3) y2+x-(a c+2 c)3
```

Po przyjęciu wyrażen program przekształca je zgodnie ze swoim zwyczajem do równań zbliżonych w zapisie do używanego przez nas sposobu notacji.

2.2.2 Przypisywanie wartości zmiennym

Operacje przypisywania wartości zmiennej realizuje operator `:`. Znak równości w programie wxMaxima odpowiada relacji porównania wartości.

```

[ (%i1) 2*f+3*kr;
  (%o1) 3 kr+2 f

```

```

[ (%i2) kr : 2;
  f : 1;
  (%o2) 2
  (%o3) 1

```

```

[ (%i4) 2*f+3*kr;
  (%o4) 8

```

```

[ (%i5) b = 3;
  (%o5) b=3

```

```

[ (%i6) b + 2;
  (%o6) b+2

```

```

[ (%i7) b : 3;
  (%o7) 3

```

```

[ (%i8) b + 2;
  (%o8) 5

```

Powyższy przykład pokazuje, że po zadeklarowaniu wartości zmiennej program będzie zastępował ją przez tę wartość zawsze, gdy zostanie ona wywołana.

2.2.3 Definiowanie i rozwiązywanie równań

Równania mogą funkcjonować w programie podobnie jak zmienne: w formie zastępujących je symboli literowych:

```

[ (%i1) Row1 : x^2 = 4;
  (%o1) x^2=4

```

```

[ (%i2) solve(Row1, x);
  (%o2) [x=-2, x=2]

```

W powyższym przykładzie zostało użyte polecenie `solve`, które wywołuje algorytm rozwiązywania równań. Jeżeli równania zgrupujemy nawiasami kwadratowymi, możemy rozwiązywać je połączone spójnikiem „i” (układy równań).

```

[ (%i1) solve([2*x+y = 4, y - x = 1], [x,y]);
  (%o1) [[x=1, y=2]]

```

Jeżeli w wyrażeniu znajdującym się na pierwszej pozycji w poleceniu `solve` nie będzie znaku równości, $F(x)$. Na przykład `solve(3*x - 6, x);`. `wxMaxima` potraktuje je jako równanie zapisane w formie ogólnej $F(x) = 0$.

Polecenie `solve` zwraca w efekcie *listę* rozwiązań nawet jeżeli istnieje tylko jedno rozwiązanie. Powoduje to pewną niedogodność, uniemożliwiającą bezpośrednie odnoszenie się do wartości rozwiązania. Można to jednak uzyskać korzystając z polecenia `ev`.

```
(%i1) rozw : solve(sin(x) = 0, x);
solve: using arc-trig functions to get a solution.
Some solutions will be lost.
(%o1) [x=0]

(%i2) 2 + solve(sin(x) = 0, x);
solve: using arc-trig functions to get a solution.
Some solutions will be lost.
(%o2) [x+2=2]

(%i3) 2 + ev(x, rozw[1]);
(%o3) 2

(%i4) 2 + ev(x + 1, rozw[1]);
(%o4) 3
```

W przytoczonym powyżej przykładzie, widać, że sama funkcja `solve` zwraca bezużyteczne, dla dalszych obliczeń, wyniki. Jednak użycie funkcji `ev` pozwala bezpośrednio wykorzystać wartość wyliczoną z równania. Oczywiście wyrażenie na pierwszym miejscu polecenia `ev` może już realizować część obliczeń, co zostało zaprezentowane w poleceniu `%i4`, gdzie wartość `x` wyznaczona z równania jest zwiększona o 1.

Po raz pierwszy spotykamy się tu z pojęciem *listy*. W programie MAXIMA przez *listę* rozumiemy skończony ciąg wartości. Do elementów *listy* odwołujemy się przez użycie indeksu tego elementu, w tym wypadku `rozw[1]`. Indeks umieszczany jest w nawiasach kwadratowych, w przypadku równania z jednym rozwiązaniem oczywistym jest, że odwołujemy się do pierwszego elementu listy, który jednocześnie jest elementem jedynym.

3 Definicja równania różniczkowego zwyczajnego

3.1 Definicja

Równaniem różniczkowym zwyczajnym rzędu n nazywamy równanie postaci:

$$F(x, y(x), y'(x), y''(x), \dots, y^{(n)}(x)) = 0, \quad (3.1)$$

gdzie poszukiwaną wartością jest funkcja: $y : [x_0, x_0 + \alpha] \rightarrow \mathbb{R}^d$, spełniająca warunek (3.1). Przy czym, $F : [x, x + \alpha] \times \underbrace{\mathbb{R}^d \times \dots \times \mathbb{R}^d}_{n+1} \rightarrow \mathbb{R}^k$ jest funkcją przynajmniej ciągłą.

Jeżeli $k=d$ oraz funkcję F można rozwikłać dla ostatniej współrzędnej, to równanie (3.1) przyjmuje postać:

$$y^{(n)}(x) = f(x, y(x), y'(x), \dots, y^{(n-1)}(x)), \quad (3.2)$$

gdzie $f : [x_0, x_0 + \alpha] \times \underbrace{\mathfrak{R}^d \times \dots \times \mathfrak{R}^d}_n \rightarrow \mathfrak{R}^d$.

Postać równania (3.1) nazywamy *uwikłaną*, natomiast równanie w postaci (3.2) nazywamy równaniem w postaci *rozwikłanej* względem pochodnej $y^{(n)}(x)$.

Zmienna y jest poszukiwaną funkcją, zmienną x nazywamy zmienną niezależną. Podczas opisu procesów fizycznych częstokroć stosujemy trochę inną nomenklaturę. W przypadku opisu ruchu (zmiana położenia w czasie) często funkcję nazywamy x , a zmienną niezależną przez t . Jest tak ze względu na tradycyjne rozumienie pewnych zmiennych wynikające z takich, a nie innych formuł definiujących pewne wielkości. Dlatego w przypadku równania różniczkowego zawsze należy się zastanowić, co tak naprawdę jest poszukiwaną funkcją a co zmienną niezależną. Taką wiedzę łatwo jest zdobyć obserwując pochodną występującą w równaniu. Wielkość, względem której różniczkujemy jest zmienną niezależną.

3.1.1 Warunki początkowe i brzegowe

Równanie różniczkowe (3.1) opisuje rodzinę funkcji, które je spełniają. Funkcji takich może być nieskończenie wiele. Aby ograniczyć liczbę rozwiązań do dokładnie jednego wprowadzamy warunki początkowe (lub brzegowe). Warunek początkowy dla równania różniczkowego rzędu n wygląda w następujący sposób:

$$\begin{cases} y(x_0) = y_0 \\ y'(x_0) = y_1 \\ \vdots \\ y^{(n-1)}(x_0) = y_{n-1} \end{cases} \quad (3.3)$$

3.1.1.1 Rozwiązanie ogólne i szczególne

Rozwiązanie równania różniczkowego zwyczajnego bez warunków brzegowych/początkowych nazywamy *rozwiązaniem ogólnym*. Po uwzględnieniu warunków brzegowych/początkowych otrzymujemy *rozwiązanie szczególne*.

4 Równania różniczkowe zwyczajne rzędu pierwszego i drugiego

4.1 Postaci równań

4.1.1 Równania w postaciach rozwikłanych

Równanie różniczkowe rzędu pierwszego w postaci rozwikłanej wygląda następująco:

$$y'(x) = f(x, y(x)). \quad (4.1)$$

Równanie różniczkowe rzędu drugiego w postaci rozwikłanej wygląda następująco:

$$y''(x) = f(x, y(x), y'(x)). \quad (4.2)$$

4.1.2 Zagadnienia początkowe (*zagadnienie Cauchy'ego*)

Zagadnienie początkowe często zwane jest też zagadnieniem Cauchy'ego. W przypadku równania różniczkowego rzędu pierwszego wygląda ono następująco:

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}, \quad (4.3)$$

dla równań różniczkowych rzędu drugiego:

$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \\ y(x_0) = y_0 \\ y'(x_0) = y_1 \end{cases}. \quad (4.4)$$

Wymienione powyżej warunki początkowe fizycznie realizują sytuację kiedy znamy stan początkowy procesu – zagadnienie (4.3) – lub stan początkowy i szybkość z jaką zmienia się badany parametr y – zagadnienie (4.4).

4.1.3 Zagadnienie brzegowe dla równania różniczkowego rzędu drugiego

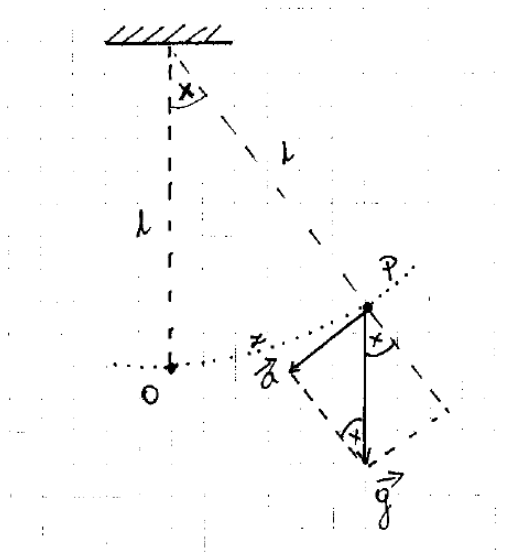
W przypadku równań różniczkowych zwyczajnych rzędu drugiego jest sens mówić o zagadnieniach brzegowych. Ma to miejsce wówczas, gdy badany stan y jest znany na początku i końcu procesu, a poszukujemy przebiegu zmian tego parametru znanymi stanami. Zagadnienie brzegowe dla równania różniczkowego zwyczajnego rzędu drugiego ma postać:

$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \\ y(x_0) = y_0 \\ y'(x_0) = y_1 \end{cases}. \quad (4.5)$$

4.2 Przykłady równań różniczkowych

4.2.1 Wahadło matematyczne

Zadanie polega na opisie ruchu wahadła matematycznego (patrz §5.1.2). Skorzystamy przy tym z praw dynamiki. Zakładamy, że wahania są niewielkie. Oznaczenia do rozwiązywanego zadania przyjmujemy takie jak na rysunku, Rysunek 4-1.



Rysunek 4-1. Schematyczna reprezentacja wahadła matematycznego omawianego w zadaniu oraz rozkład sił.

P – jest punktem w jakim znajduje się punkt materialny o masie m ;

a – jest przechłodziem wahadła;

g – jest przyspieszeniem ziemskim;

z – jest odchyleniem od środka O mierzonym po łuku OP

Siła grawitacji działająca na punkt m w P rozkłada się na dwie składowe: siłę równoważoną przez naciąg nici oraz siłę, od której pochodzi przyspieszenie a , styczną do łuku OP w punkcie P i skierowaną do punktu O . Siły działające na punkt materialny tworzą trójkąt prostokątny, z którego można zapisać bilans sił działających na punkt P .

$$m\vec{a} = -m\vec{g} \sin(x) \quad (4.6)$$

we wzorze (4.6) występuje znak minus, gdyż jest to siła działająca ze zwrotem przeciwnym do ruchu. Przyspieszenie jest drugą pochodną drogi względem czasu.

$$a = \frac{d^2 z}{dt^2} = z \quad (4.7)$$

$$m \frac{d^2 z}{dt^2} = -mg \sin(x) \quad (4.8)$$

z definicji radiana (patrz §5.1.1): $z = xl$, skoro $l = const.$: $\frac{d^2 z}{dt^2} = l \frac{d^2 x}{dt^2}$, wstawiając co równania, po odpowiednich przekształceniach otrzymujemy:

$$\frac{d^2 x}{dt^2} = -\frac{g}{l} \sin(x) \quad (4.9)$$

dla bardzo małych odchyłeń można przyjąć $\sin(x) = x$ natomiast iloraz g i l można zastąpić stałą związaną z własnościami wahadła, np. α . Wówczas równanie upraszcza się do:

$$\frac{d^2 x}{dt^2} = -\alpha x. \quad (4.10)$$

Równanie (4.10) opisuje ruch wahadła matematycznego przy założeniu niewielkich wychyleń. Jest to równanie różniczkowe zwyczajne drugiego rzędu. Równania tego typu możemy rozwiązywać dokładnie w programie MAXIMA.

Zgodnie z tym, co zostało powiedziane wcześniej, rozwiązanie ogólne będzie zależało od dwóch parametrów. Odpowiada to różnym przypadkom ruchu tego wahadła. Dlatego też, aby otrzymać jednoznaczne rozwiązanie w postaci zależności $x(t)$, należy dodać warunki początkowe lub brzegowe:

1. *Warunki początkowe* – zakładamy, że w momencie rozpoczęcia obliczeń znamy kąt o jaki wychylone jest wahadło oraz jego prędkość początkową:

$$\begin{cases} \frac{d^2 x}{dt^2} = -\alpha x \\ x(t_0) = x_0 \\ x'(t_0) = x_1 \end{cases}$$

2. *Warunki brzegowe* – przyjmujemy, że analizujemy przebieg drgań wahadła, w którym znane są kąty x_a, x_b o jakie wychyla się ono w chwilach t_a i t_b . Na podstawie tych danych możemy poznać przebieg procesu w przedziale czasu

$$[t_a, t_b], \text{ a nawet dalej: } \begin{cases} \frac{d^2 x}{dt^2} = -\alpha x \\ x(t_a) = x_a \\ x(t_b) = x_b \end{cases}$$

4.2.2 Poszukiwanie rozwiązania dokładnego problemu drgań wahadła matematycznego

Otrzymane w poprzednim paragrafie (§4.2.1) równania różniczkowe wraz z warunkami początkowymi/brzegowymi można rozwiązać analitycznie w programie wxMaxima. Również tutaj nie ma znaczenia wybór nazw zmiennych, to czy mamy do czynienia z poszukiwaną funkcją, czy zmienną niezależną zależy od ulokowania symbolu w formule.

4.2.2.1 Rozwiązanie ogólne

Dla uproszczenia przyjmijmy, że długość nici wynosi $l = l$ [m], a przyspieszenie ziemskie jest równe w przybliżeniu $g = 9.81$ [m s⁻²]. Oto jak rozwiązujemy takie zagadnienie w wxMaximie:

```

(%i1) RRZ1 : 'diff(x, t, 2) = -9.81*x;
(%o1)  $\frac{d^2}{dt^2} x = -9.810000000000001 x$ 

(%i2) ode2(RRZ1, x, t);
rat: replaced 9.810000000000001 by 981/100 = 9.810000000000001
(%o2)  $x = \%k1 \sin\left(\frac{3\sqrt{109} t}{10}\right) + \%k2 \cos\left(\frac{3\sqrt{109} t}{10}\right)$ 

```

W powyższym zapisie 'diff oznacza operator „gotowość” do różniczkowania. W przeciwieństwie do operatora diff nie wykonuje on różniczkowania, ale przygotowuje pochodną, która później jest elementem równania różniczkowego. Operatory diff i 'diff mają następującą składnię:

diff(wyrażenie, zmienna, stopień_pochodnej) – wykonuje różniczkowanie symboliczne wyrażenia wyrażenie względem zmiennej zmienna stopień_pochodnej razy. Jeżeli stopień_pochodnej nie jest zdefiniowany przyjmowana jest wartość 1.

'diff(funkcja, zmienna stopień_pochodnej) – wprowadza symbol pochodnej (nie wykonuje różniczkowania) stopień_pochodnej stopnia względem zmiennej zmienna z funkcji funkcja.

```

(%i1) diff(x^5+x^2+x+1, x);
(%o1)  $5 x^4 + 2 x + 1$ 

(%i2) diff(x^5+x^2+x+1, x, 2);
(%o2)  $20 x^3 + 2$ 

(%i3) diff(x^5+x^2+x+1, x, 4);
(%o3)  $120 x$ 

(%i4) 'diff(y, x, 3);
(%o4)  $\frac{d^3}{dx^3} y$ 

(%i5) 'diff(x, t);
(%o5)  $\frac{d}{dt} x$ 

(%i6) 'diff(2*x-4, x, 2);
(%o6)  $\frac{d^2}{dx^2} (2 x - 4)$ 

```

Wartości zmiennoprzecinkowe wprowadzane do komputera nie zawsze mogą być reprezentowane dokładnie. Komputer liczy na skończonym zbiorze liczb

zmiennoprzecinkowych i dokładnie może wyrazić tylko liczby z tego zbioru. Niestety jak pamiętamy pomiędzy każdymi dwiema liczbami wymiernymi można wskazać liczbę wymierną. Arytmetyka komputerowa na to nie pozwala. Zatem przy pomocy liczb zmiennoprzecinkowych nie jesteśmy w stanie wyrazić nawet całego zbioru liczb wymiernych, nie mówiąc o liczbach rzeczywistych. MAXIMA wykonuje obliczenia symbolicznie, dlatego możemy dostrzec, że wartość przyspieszenia ziemskiego zostanie przekształcona do wyrażenia $981/100$. Co poprawia dokładność obliczeń, jako że nie są one przeprowadzane dla wartości dziesiętnych, a dla związanych ze sobą wartości całkowitych (które mogą być wyrażone dokładnie). Program informuje o tym użytkownika informacją:

```
rat : replaced 9.8100000000000001 by 981/100 = 9.8100000000000001
```

W rozwiązaniu ogólnym równania różniczkowego zwyczajnego występują stałe, oznaczone %k1 i %k2. W zależności od warunków początkowych/brzegowych wartości te wyrażone są przez wartości rzeczywiste.

4.2.2.2 Rozwiązania szczególne

Rozwiązania szczególne wyznaczamy w MAXIMIE korzystając z poleceń ic2 i bc2.

```
(%i1) RRZ1 : 'diff(x, t, 2) = -9.81*x;
(%o1)  $\frac{d^2}{dt^2} x = -9.8100000000000001 x$ 

(%i2) ode2(RRZ1, x, t);
rat: replaced 9.8100000000000001 by 981/100 = 9.8100000000000001
(%o2)  $x = \%k1 \sin\left(\frac{3\sqrt{109} t}{10}\right) + \%k2 \cos\left(\frac{3\sqrt{109} t}{10}\right)$ 

(%i3) ic2(%o2, t=0, x=0.5, 'diff(x,t) = 0);
rat: replaced 0.5 by 1/2 = 0.5
(%o3)  $x = \frac{\cos\left(\frac{3\sqrt{109} t}{10}\right)}{2}$ 

(%i4) bc2(%o2, t=0, x=0.5, t=1, x=0.25);
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.25 by 1/4 = 0.25
(%o4)  $x = \frac{\cos\left(\frac{3\sqrt{109} t}{10}\right) \left(2 \cos\left(\frac{3\sqrt{109}}{10}\right) - 1\right) \sin\left(\frac{3\sqrt{109} t}{10}\right)}{4 \sin\left(\frac{3\sqrt{109}}{10}\right)}$ 
```

Wyniki obliczeń można zaznaczyć przy pomocy wskaźnika myszy i skopiować do schowka, a następnie wykorzystać podczas dalszych obliczeń.

4.2.3 Prezentacja uzyskanych wyników

Otrzymany wynik jest pewną funkcją, określoną przedziale $[t_0, t_0 + \alpha]$. Funkcję tę możemy wykreślić w programie MAXIMA. Aby tego dokonać należy najpierw przypisać jej

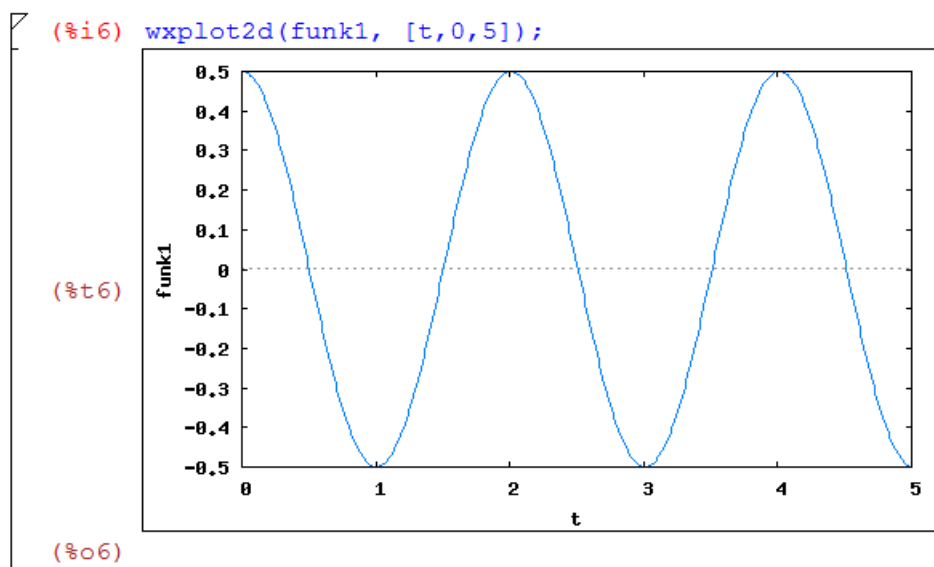
pewną nazwę. Wykonujemy, to korzystając z wymienionej wcześniej możliwości kopiowania do schowka treści zwróconej przez system.

```
(%i5) funk1(t) := cos((3*sqrt(109)*t)/10)/2;
```

$$\cos\left(\frac{3\sqrt{109}t}{10}\right)$$

```
(%o5) funk1(t) := \frac{\cos\left(\frac{3\sqrt{109}t}{10}\right)}{2}
```

Następnie wykres tej funkcji możemy narysować w odpowiednim przedziale. Ze względu na matematyczne podstawy teorii rozwiązywania równań różniczkowych zwyczajnych, przedział, w którym rysujemy wykres funkcji powinien rozpoczynać się od wartości t_0 . Do rysowania wykresu funkcji zależnej od jednego argumentu wykorzystujemy funkcję `plot2d` (wywołuje zewnętrzny program **gnuplot**) lub jej wersję do umieszczania wykresu w arkuszu `wxplot2d`.



Na osi pionowej odczytujemy kąt o jaki wychyla się wahadło w pewnej chwili czasu, którą możemy odczytać z osi poziomej.

4.2.3.1 Więcej funkcji na jednym wykresie

Możliwe jest też umieszczanie większej ilości funkcji na jednym wykresie oraz kontrolowanie pewnych parametrów wykresu. Tutaj poprzez przykład zostanie zaprezentowana ingerencja w niektóre z opcji kreślenia wykresów.

Spróbujemy narysować wykres funkcji oraz jej pochodnej, opisać osie i krzywe występujące na wykresie.

```
(%i5) funkl(t) := cos((3*sqrt(109)*t)/10)/2;
```

$$(\%o5) \text{ funkl}(t) := \frac{\cos\left(\frac{3\sqrt{109}t}{10}\right)}{2}$$

```
(%i6) diff(funkl(t), t);
```

$$(\%o6) \frac{3\sqrt{109} \sin\left(\frac{3\sqrt{109}t}{10}\right)}{20}$$

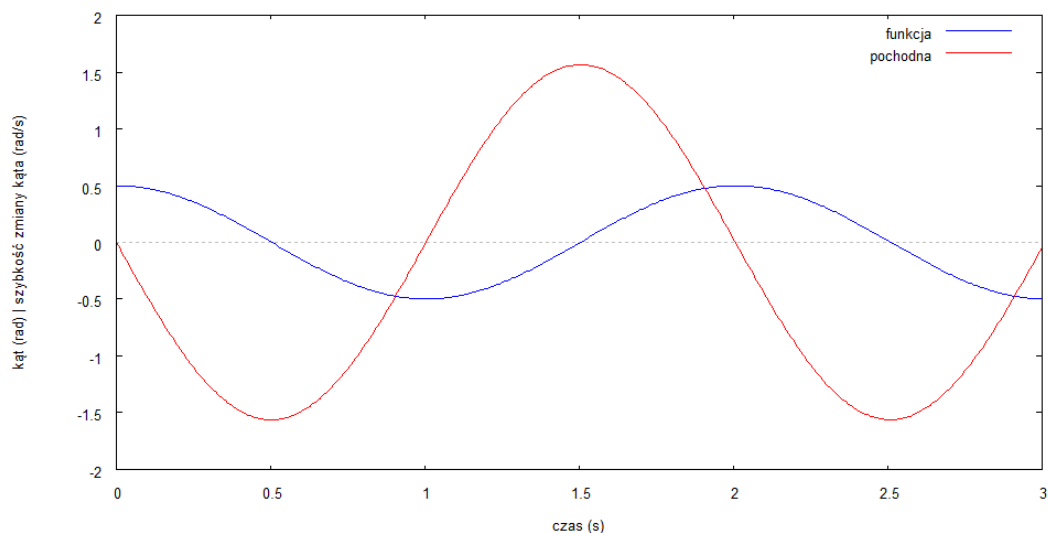
```
(%i7) pochodna1(t) := -(3*sqrt(109)*sin((3*sqrt(109)*t)/10))/20;
```

$$(\%o7) \text{ pochodna1}(t) := -\frac{3\sqrt{109} \sin\left(\frac{3\sqrt{109}t}{10}\right)}{20}$$

Powyzsze polecenia pozwalaja wyznaczyć zmienne opisujące wyznaczoną funkcję oraz jej pochodną. Aby narysować te krzywe na jednym wykresie stosujemy polecenie `plot2d`:

```
(%i8) plot2d( [funkl(t), pochodna1(t)], [t, 0, 3],
  [legend, "funkcja", "pochodna"], [xlabel, "czas (s)"],
  [ylabel, "kąt (rad) | szybkość zmiany kąta (rad/s)"] );
```

wykras otwiera się w nowym oknie:



Zastosowane tu polecenie `plot2d` zawiera dodatkowe opcje: wzięte w nawiasy kwadratowe nazwy rysowanych funkcji pozwalają umieścić więcej niż jedną krzywą na wykresie; `legend` pozwala dodać legendę, opisy w legendzie wzięte są w cudzysłowie i będą przypisane do konkretnych funkcji w takiej kolejności, w jakiej pojawiały się ich nazwy; `xlabel` i `ylabel` pozwala nadać opisy osiom x i y .

5 Schematy różnicowe dla równań różniczkowych zwyczajnych

W dalszej części będziemy zajmować się przybliżonym rozwiązywaniem równań różniczkowych zwyczajnych. Zakładamy, że podstawowy problem, z którym mamy do czynienia wygląda następująco:

$$y'(x) = f(x, y(x)) \quad (5.1)$$

Jeżeli funkcja po prawej stronie zależy tylko od zmiennej y równanie to nazywamy autonomicznym. W takim przypadku pole prędkości zmiany parametru y jest stałe w całej dziedzinie równania różniczkowego. Jeżeli f jest zależne od x , to pole prędkości zmienia się w zależności od x . Równanie różniczkowe zwyczajne nazywamy wówczas nieautonomicznym.


Rozwiązać numerycznie równanie różniczkowe znaczy to podać przybliżone wartości rozwiązania równania różniczkowego wraz z ustaloną funkcją f , warunkami początkowymi, brzegowymi w dyskretnym zbiorze punktów należących do dziedziny równania.

5.1 Obserwacja wrażliwości równania różniczkowego na wartości początkowe oraz na parametry modelu

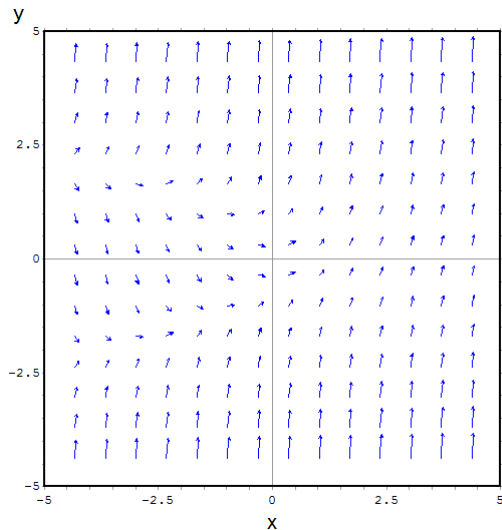
5.1.1 Pole kierunków związane z równaniem różniczkowym

Pole kierunków opisane równaniem różniczkowym (5.1) w układzie $x \times \mathfrak{R}^n$ wygląda następująco $[1, y'(x)]$, zatem $[1, f(x, y(x))]$. Pole kierunków opisanych równaniem różniczkowym, to zbiór wektorów stycznych do rozwiązania równania (5.1).

W MAXIMie istnieje możliwość przedstawienia pola kierunków. Realizuje to polecenie `plotdf`:

```
 (%i1) plotdf(y^2 + x, [x, -5, 5], [y, -5, 5]);
```

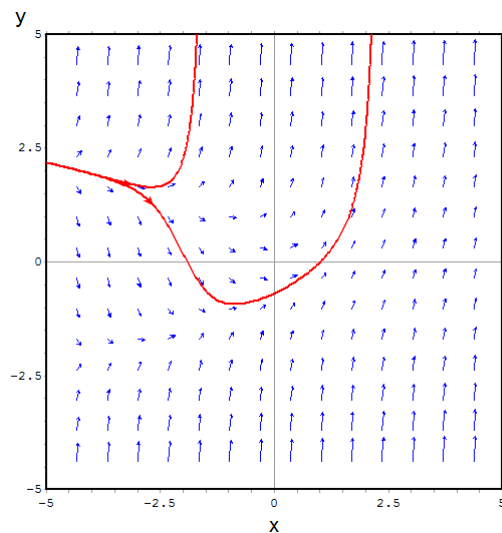
W efekcie otrzymamy następujące pole:



Strzałki na wykresie obrazują chwilowe prędkości w różnych punktach. Spróbujmy wybrać któryś z tych punktów na tym wykresie. Jak wtedy będzie się poruszał punkt startujący z wybranego punktu? Wyobraźmy sobie trajektorię jego ruchu.

5.1.2 Wrażliwość rozwiązania na wybór warunku początkowego

Odpowiadając na pytanie, które pojawiło się na końcu poprzedniego paragrafu wyobraziliśmy sobie trajektorię ruchu rozpoczynającego się w pewnym punkcie obszaru umieszczonego na wykresie. Możemy teraz sprawdzić czy nasza intuicja była poprawna. Kliknijmy wybrany przez nas punkt. Na wykresie pojawi się trajektoria. Kliknijmy jeszcze raz, pojawi się kolejna:

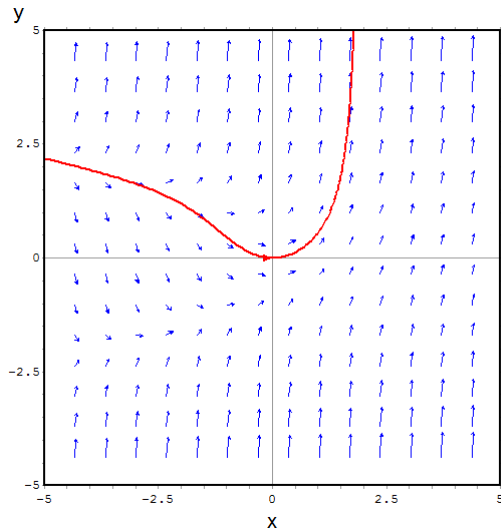


Jak widać na rysunku kształt krzywej bardzo znacząco zależy od miejsca, w którym kliknęliśmy. Zaznaczona na czerwono krzywa to rozwiązanie szczególne problemu początkowego Cauchy'ego (4.3).

Wybór punktu początkowego przez kliknięcie nie jest najdokładniejszy. Skoro krzywa jest aż tak bardzo wrażliwa na wybór warunku początkowego potrzebujemy dodatkowego narzędzia pozwalającego wybrać konkretny punkt. Wprowadzając polecenie:

```
(%i2) plotdf(y^2 + x, [x, -5,5], [y, -5,5], [trajectory_at, 0, 0]);
```

w wyniku czego otrzymujemy następujące pole kierunków wraz z narysowanym rozwiązaniem przechodzącym przez punkt $[0, 0]$:



5.1.3 Wrażliwość rozwiązania na parametry modelu

Dotychczasowa analiza pozwoliła nam zaobserwować wpływ wyboru warunku początkowego na kształt rozwiązania równania różniczkowego zwyczajnego. Spróbujmy teraz zastanowić się nad innym zagadnieniem. Funkcja f stojąca po prawej stronie równania różniczkowego (5.1) ma pewną postać. Zarówno przed zmienną x jak i y możemy wstawiać dowolne współczynniki. Czy będzie to miało duży wpływ na pole kierunków? Na kształt krzywej będącej rozwiązaniem?

Przeanalizujemy ten problem przy użyciu narzędzi dostarczonych przez program MAXIMA. Wprowadzając następujące wyrażenie:

```
(%i3) plotdf(m*y^2 + k*x, [sliders, "m=-2:3, k=-2:3"],
[x, -5,5], [y, -5,5], [trajectory_at, 0, 0]);
```

otrzymujemy pole kierunków wraz z suwakami, które pozwalają zmodyfikować wartości parametrów m i k :



Modyfikacja wartości tych parametrów pozwala obserwować jak bardzo zmienia się krzywa rozwiązania w zależności od przyjętego modelu. Zmiana wartości parametrów m i k jest związana z modyfikacją parametrów fizycznych opisywanego procesu. Znaczące zmiany w kształcie krzywej będącej rozwiązaniem problemu początkowego pokazują jak duże błędy można popełnić przez zbyt grube szacowanie, choćby parametrów termofizycznych.

5.1.4 Ćwiczenia

Proszę przeanalizować w sposób analogiczny do opisanego w §5.1.2 i §5.1.3 następujące równania różniczkowe:

- a) $y'(x) = m*y + k*x^2$
- b) $y'(x) = s*y$

5.2 Schemat Eulera

Omówione powyżej pole kierunków związanych z równaniem różniczkowym daje nam intuicję na temat sposobów tworzenia metod numerycznych.

5.2.1 Wyprowadzenie schematu przy wykorzystaniu interpretacji geometrycznej

Należy rozwiązać problem początkowy (4.3). Rozwiązanie będzie wyznaczone w dyskretnym zbiorze punktów. Dziedzinę równania dzielimy według ustalonego kroku całkowania $h = (\Delta x)$. Stojąc w punkcie o współrzędnych (x_0, y_0) poruszamy się zgodnie z kierunkiem wyznaczonym przez pole kierunków. W punkcie (x_0, y_0) wektor zdefiniowany przez równanie różniczkowe jest postaci $[1, y'(x_0)] = [1, f(x_0, y_0)]$. Poruszając się we wskazanym przez niego kierunku z krokiem o długości Δx docieramy do punktu $(x_0, y_0) + \Delta x [1, f(x_0, y_0)] = (x_0 + \Delta x, y_0 + \Delta x f(x_0, y_0))$. Chcąc obliczyć w ten sposób wartość funkcji $y(x)$ w punkcie x_k , należy wykonać N^1 kroków opisanym schematem.

Powstały w ten sposób schemat przybliżania rozwiązania równania różniczkowego można zapisać w następującej postaci:

$$\begin{cases} x_{i+1} = x_i + h \\ y_{i+1} = y_i + hf_i \end{cases} \text{ dla } i = 0, 1, \dots, N-1 \quad (5.2)$$
$$h = \Delta x, f_i = f(x_i, y_i)$$

Algorytm ten tworzy na płaszczyźnie, będącej dziedziną rozwiązania, łamaną, która powinna odtwarzać kształt krzywej będącej rozwiązaniem.

Względem na omówioną w poprzednim rozdziale §5.1.2 wrażliwość rozwiązania równania różniczkowego na wybór warunku początkowego sugeruje od razu problem z dokładnością. Tak naprawdę, w każdym kolejnym kroku schematu stajemy przed zadaniem rozwiązania innego problemu początkowego. Równanie różniczkowe jest takie samo, jednak zmienia się warunek początkowy. Prawie zawsze kolejne punkty naszej łamanej znajdują się poza krzywą opisującą rozwiązanie dokładne. Startując z punktu leżącego poza krzywą możemy już nigdy na nią nie wrócić, a czasem bardzo szybko oddalać się od niej. Oddalanie się nie będzie w tym wypadku spowodowane dodatkowymi błędami, a raczej charakterem pola kierunków w obszarze do którego przesunęliśmy się. Nawet najdrobniejsze przesunięcie się z krzywej

¹ Dla podanych w przykładzie parametrów $N = \frac{(x_k - x_0)}{\Delta x}$

rozwiązania dokładnego powoduje, że trafiamy na krzywą będącą rozwiązaniem problemu o innym warunku początkowym.

Jednak pomimo wymienionych niedoskonałości, ze względu na bardzo intuicyjny charakter, prostotę implementacji oraz nieduży koszt obliczeń metoda ta jest bardzo szeroko stosowana. Nawet w przypadku bardzo skomplikowanych zagadnień osoby, które próbują znaleźć ich rozwiązania numeryczne najpierw próbują tego dokonać metodą Eulera. Dopiero kiedy ona zawodzi podejmowane są próby zastosowania bardziej złożonych schematów.

5.2.2 Przykład obliczeń schematem Eulera

W tym paragrafie podjęta zostanie próba znalezienia przybliżonego rozwiązania zagadnienia początkowego Cauchy'ego:

$$\begin{cases} y' = 0.6x^2 - 1.6y \\ y(0) = 3 \end{cases} \quad (5.3)$$

Rozwiązanie dokładne tego problemu możemy znaleźć przy pomocy programu MAXIMA, wykonując polecenia:

```
ode2('diff(y,x) = 0.6*x**2 - 1.6*y, y, x);
ic1(%, x=0, y=3);
```

Rozwiązanie dokładne jest postaci:

$$y = \frac{e^{-\frac{8x}{5}} \left((96x^2 - 120x + 75) e^{\frac{8x}{5}} + 693 \right)}{256}$$

Wyznaczoną zależność przejmujemy przy użyciu myszy i korzystając z operacji kopiuj do schowka oraz wklej definiujemy funkcję

```
rozDokl(x) := (e^(-(8*x)/5) * ((96*x^2 - 120*x + 75)
    * e^((8*x)/5) + 693)) / 256;
```

Jak widać uzyskane rozwiązanie jest bardzo skomplikowane i być może znalezienie go stanowiłoby problem dla niewprawnej osoby. Teraz zastosujemy metodę Eulera do tego zadania.

Odtworzenie rozwiązania równania różniczkowego wymaga przechowywania wszystkich punktów łamanej, która je przybliża. Jest ona zdefiniowana przez dwa ciągi punktów x_i, y_i . Można oczywiście za każdym razem definiować nową zmienną, w której przechowujemy kolejne przybliżenie, jednak podejście takie jest mało efektywne, gdyż uniemożliwia automatyzację obliczeń. Innym podejściem jest praca na tablicach. W ten sposób zapewniona jest zawsze taka sama nazwa zmiennej, a odwołanie do kolejnego elementu tablicy odbywa się przez wybór odpowiedniego indeksu.

```

(%i1) xi : make_array(flonum, 11);
      yi : make_array(flonum, 11);
(%o1) Lisp array [11]
(%o2) Lisp array [11]

```

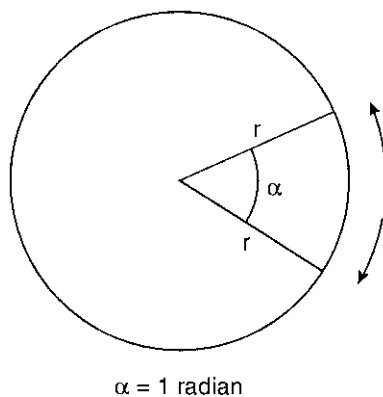
W MAXIMIE istnieje wiele sposobów definiowania tablic. W tym przykładzie wykorzystano polecenie `make_array`. Jako argumenty tego polecenia podajemy typ elementów tablicy, w tym wypadku `flonum` – typ zmiennoprzecinkowy i liczbę elementów w tablicy – tablica numerowana jest indeksami od 0 do N-1 (N liczba podana jako drugi argument polecenia `make_array`)

6 Dodatek A

6.1 Definicje

6.1.1 Radian

Jest to miara łukowa kąta płaskiego, jest jednostką uzupełniającą układu SI. Przez jeden radian rozumiemy kąt płaski wycinający z okręgu łuk równy promieniowi tego okręgu (patrz Rysunek 6-1. Graficzne przedstawienie kąta o mierze jednego radiana). Radian [rad] jest jednostką układu SI. Jest jednostką niemianowaną.



Rysunek 6-1. Graficzne przedstawienie kąta o mierze jednego radiana

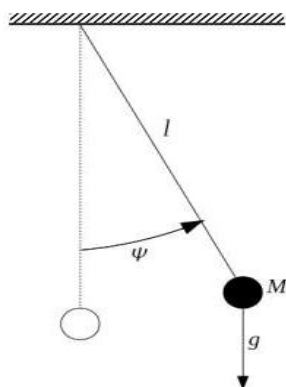
Korzystając z definicji radiana można wyznaczyć wartość dowolnego kąta. Jeżeli kąt o mierze α wycina z okręgu o promieniu R łuk o długości l , jego miarę w radianach znajdziemy stosując wzór:

$$\alpha = \frac{l}{R} [\text{rad}]. \quad (6.1)$$

Jeden radian w mierze stopniowej jest równy $57,29578^\circ$. Kąt pełny w radianach jest równy 2π .

6.1.2 Wahadło matematyczne

Wahadłem matematycznym nazywamy punkt materialny o masie M zawieszony na długiej, cienkiej i nierozciągliwej nici, Rysunek 6-2. Wahadło matematyczne Wykonuje ono wahania wokół najniższego położonego punktu O , zwanego środkiem wahań.



6-2. Wahadło matematyczne

6.1.3 Wzór Taylora i rozwinięcie funkcji w szereg potęgowy

6.1.3.1 Wzór Taylora

Założmy, że $f: [a, b] \rightarrow Y$ jest $(n+1)$ razy różniczkowalną funkcją w sposób ciągły. Wówczas dla każdego $x \in (a, b)$:

$$f(x) = f(a) + \frac{(x-a)}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \dots + \frac{(x-a)^n}{n!} f^{(n)}(a) + R_n(x, a) \quad (6.2)$$

gdzie $R_n(x, a)$ spełnia: $\lim_{x \rightarrow a} \frac{R_n(x, a)}{(x-a)^n} = 0$.

6.1.3.2 Szereg Taylora

Niech funkcja $f(x)$ ma w pewnym otoczeniu punktu x_0 wszystkie pochodne i niech $x_0 + h$ należy do tego otoczenia. W myśl wzoru Taylora mamy dla $n=1, 2, \dots$

$$f(x_0 + h) = f(x_0) + f'(x_0) \frac{h}{1!} + \dots + f^{(n-1)}(x_0) \frac{h^{n-1}}{(n-1)!} + R_n(h) \quad (6.3)$$

gdzie:

$$R_n(h) = f^{(n)}(\xi) \frac{h^n}{n!}, \quad \xi \in (x_0, x_0 + h) \quad (6.4)$$

6.1.3.3 Szereg Maclaurina