

Metody numeryczne

Wykład 1



Dr inż. Michał Łanczont

Instytut Elektrotechniki i Elektrotechnologii

E419, tel. 4293, m.lanczont@pollub.pl, <http://lanczont.pollub.pl>

Informacje wstępne - ED

Wykład – 2h co dwa tygodnie

Wykład zaliczany jest na podstawie ocen z kolokwiów (sprawdzianów - testów).

Laboratorium – 2h co dwa tygodnie

Indywidualna decyzja prowadzącego daną grupę laboratoryjną – oceny z kartkówek, sprawozdań, odpowiedzi...

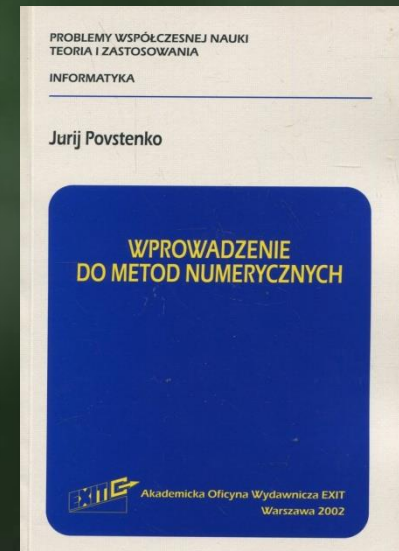
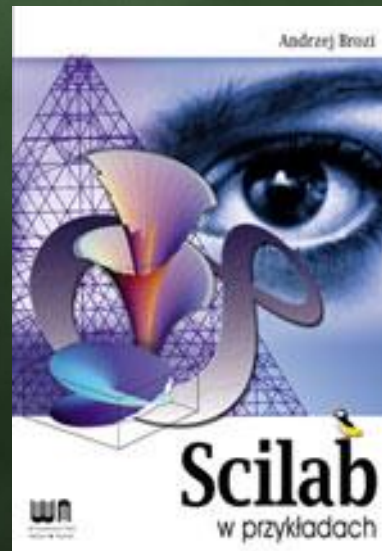
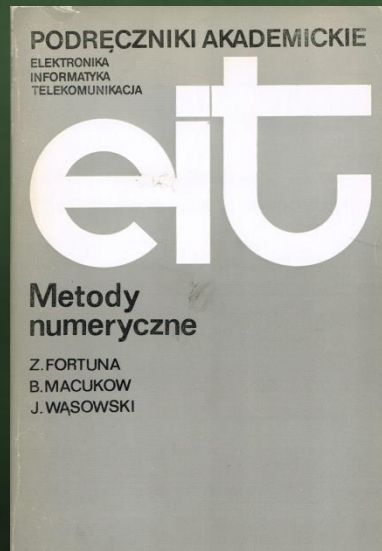
Literatura

D. Kincaid, W. Chaney: **Analiza numeryczna**, WNT, Warszawa, 2006

Z. Fortuna, B. Macukow, J. Wąsowski: **Metody numeryczne**, WNT, Warszawa, 1998

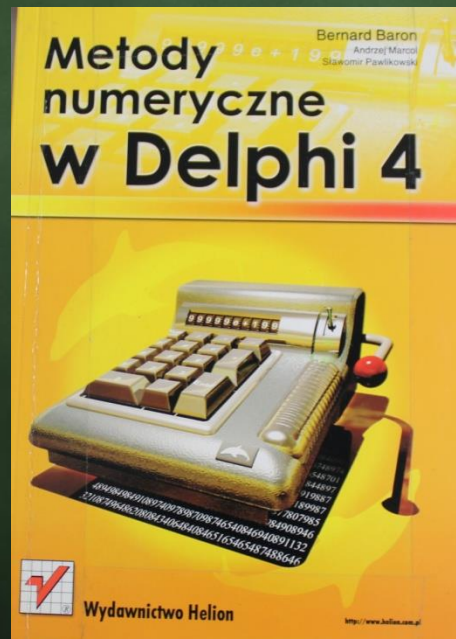
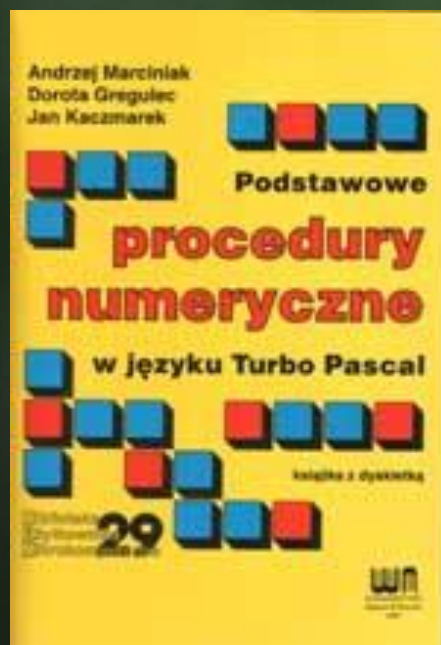
A. Brozi: **Scilab w przykładach**, Nakom, Poznań, 2007

J. Povstenko: **Wprowadzenie do metod numerycznych**, EXIT, Warszawa, 2002



Literatura

- A. Marciniak, D. Gregulec, J. Kaczmarek: **Podstawowe procedury numeryczne w języku Turbo Pascal**, Nakom, Poznań, 2000
- B. Baron, A. Marcol, S. Pawlikowski: **Metody numeryczne w Delphi 4**, Helion, Gliwice, 1999
- L. Jaroszyński, M. Łanczont: **Laboratorium Metod Numerycznych**, Wydawnictwo Politechniki Lubelskiej, Lublin, 2014



Zakres materiału

1. Pojęcia podstawowe – równanie Taylora, rząd zbieżności, równania różnicowe
2. Arytmetyka komputerowa - błędy, stabilność i niestabilność algorytmów
3. Rachunek macierzowy
4. Równania liniowe
5. Interpolacja i aproksymacja
6. Równania nieliniowe
7. Różniczkowanie i całkowanie numeryczne
8. Równania różniczkowe

Zakres wykładu

- Pojęcia podstawowe
- Arytmetyka zmiennopozycyjna
- Błędy obliczeń numerycznych
- Szacowanie błędów
- Źródła błędów
- Przenoszenie błędów
- Utrata cyfr znaczących
- Uwarunkowania zadań
- Stabilność algorytmów

Środowisko programowania

Języki programowania

Fortran

C, Pascal, C++, ...

Środowiska obliczeniowe

Matlab

Scilab

OpenModelica

Mathematica

Scilab

www.scilab.org

Aktualna wersja **6.1.0**

Windows (32/64), Linux (32/64)

MacOSX (64)

Obliczenia matematyczne i symulacje

Wykresy 2D i 3D

Statystyka, Optymalizacja

Analiza sygnałów

Xcos – odpowiednik Simulink (Matlab)

Scilab Home Page | www.scilab.org

https://www.scilab.org/ Szukaj

AstroCenter · AstroPri... Setup

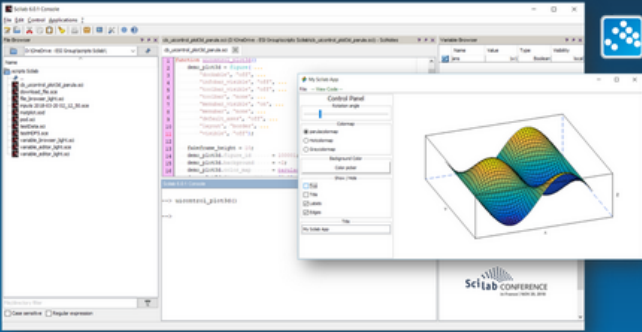
Scilab

Download Software ▾ Tutorials Applications ▾ Services ▾ Cloud ▾ About ▾

Search

Download Scilab 6.1.0
Windows, Linux and Mac OS X

Open source software for numerical computation



Discover Scilab

- [What is Scilab ?](#)
- [Features](#)
- [Xcos](#)

Scilab versions

- [Scilab 6.1.0](#)
- [Scilab 6.0.2](#)
- [Scilab 5.5.2](#)
- [System requirements](#)

News

- [Internet Of Things with Scilab 6.1](#)
- [Web tools in Scilab 6.1](#)

[more news...](#)

Scilab 6.1.0 | www.scilab.org

https://www.scilab.org/download/6.1.0

AstroCenter · AstroPri... Setup

Scilab

Download Software ▾ Tutorials Applications ▾ Services ▾ Cloud ▾ About ▾

Home ▸ Download

Scilab 6.1.0

Software versions

- [Scilab 6.1.0](#)
- [Scilab 6.0.2](#)
- [Scilab 5.5.2](#)
- [Scilab](#)
- [Previous versions](#)
- [Latest version](#)
- [System Requirements](#)
- [To donate](#)

Released on Tue, 25 Feb 2020

[System requirements](#) | [Change log](#)

Scilab 6.1.0 is released under the terms of the GNU General Public License (GPL) v2.0. Prior to version 6.0.0, Scilab was licensed under the terms of the CeCILL license v2.1, and continues to be available under such terms.

Windows Vista, 7, 8, 10

- [Scilab 6.1.0 - Windows 64 bits \(exe\)](#)
- [Scilab 6.1.0 - Windows 32 bits \(exe\)](#)

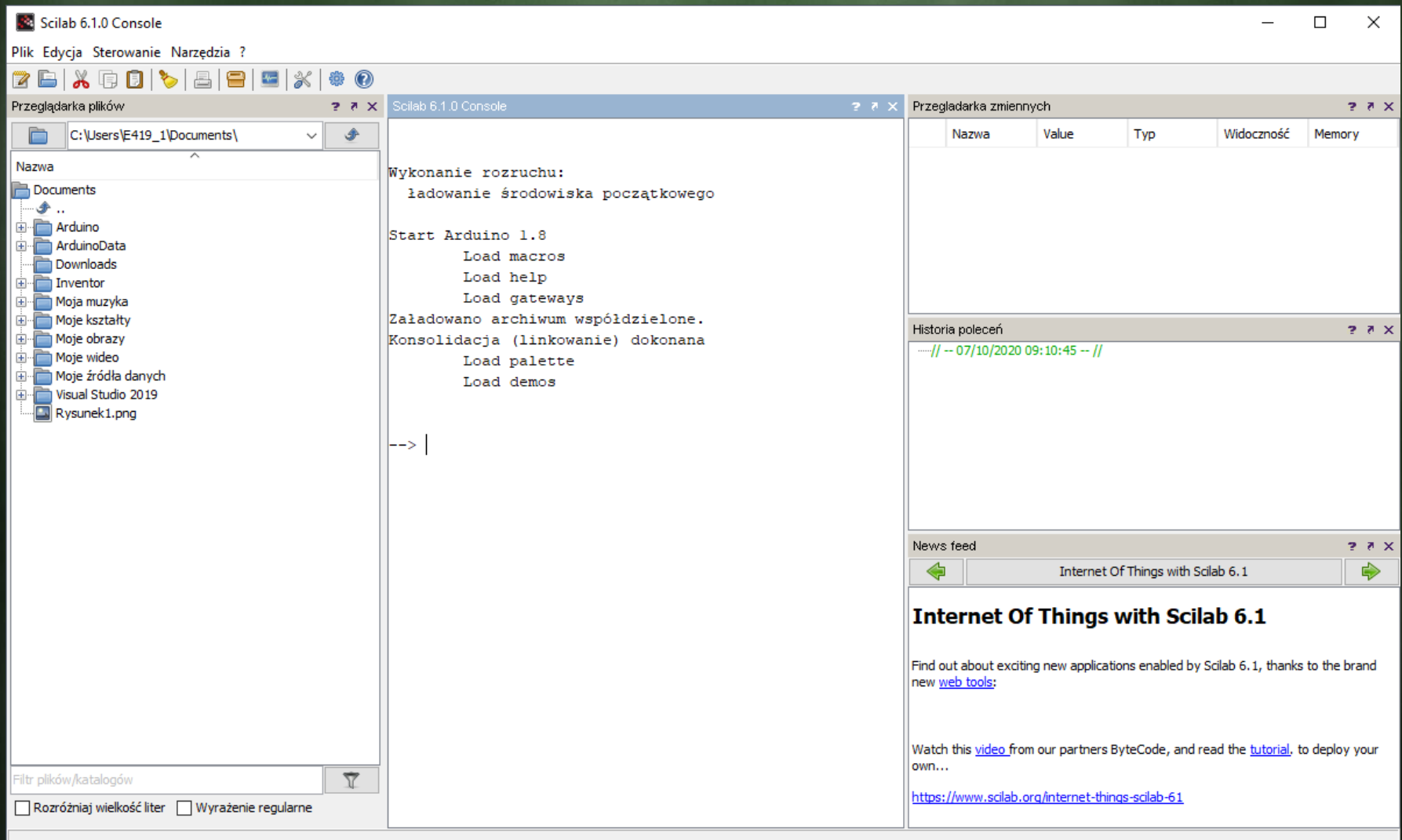
Good chances are, you have a 64 bits machine!
But the 32 bits version will work in any cases.

GNU/Linux

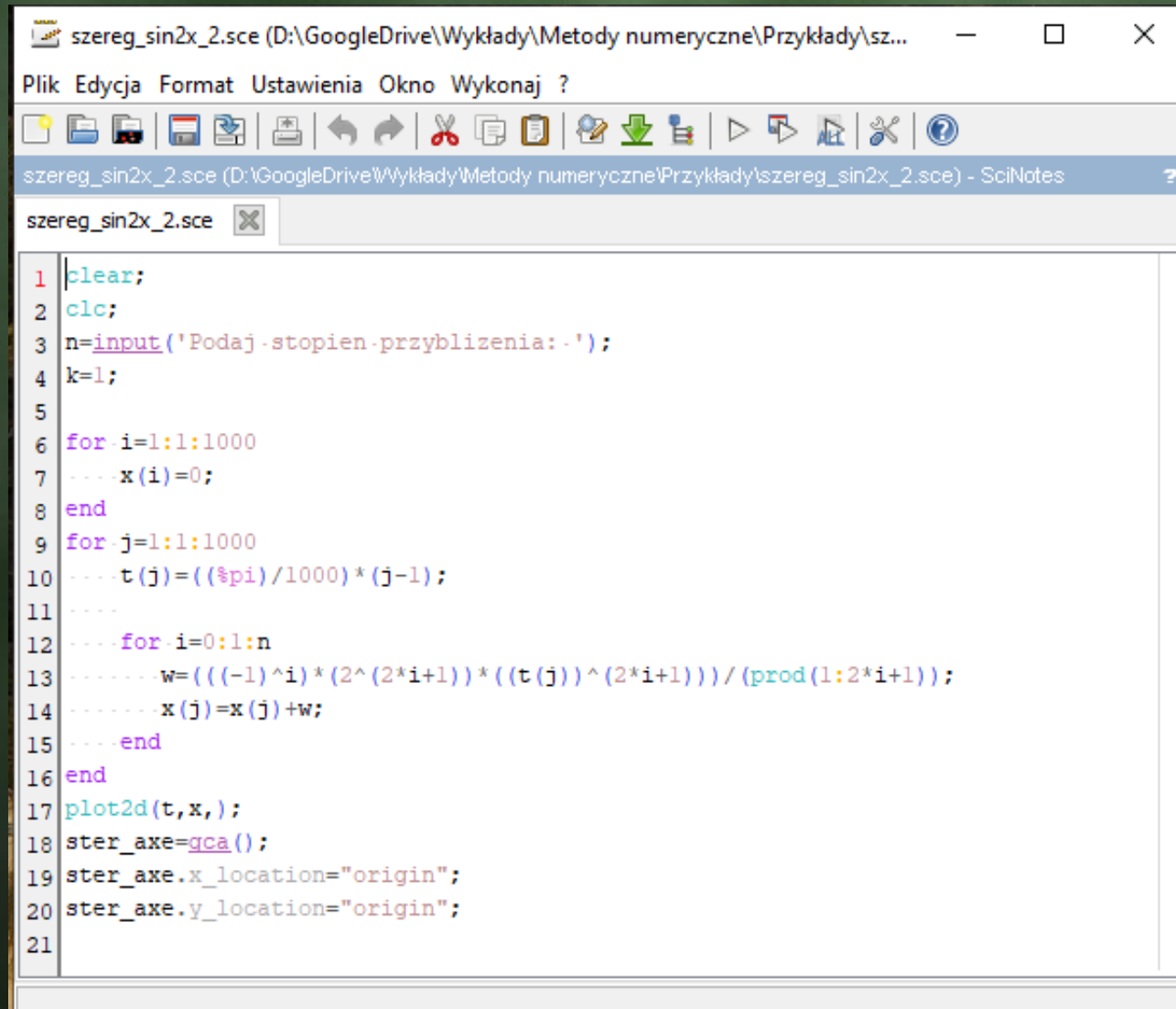
- [Scilab 6.1.0 - Linux 64 bits, \(scilab-6.1.0.bin.linux-x86_64.tar.gz - 241M\) \(gz\)](#)
- [Scilab 6.1.0 - Linux 32 bits, \(scilab-6.1.0.bin.linux-i686.tar.gz - 244M\) \(gz\)](#)

Mac OS X

- [Scilab 6.1 branch - MacOSX 64 bits, \(scilab-6.0.2-x86_64.dmg - 295M\) \(dmg\)](#)



Scilab - Scinotes

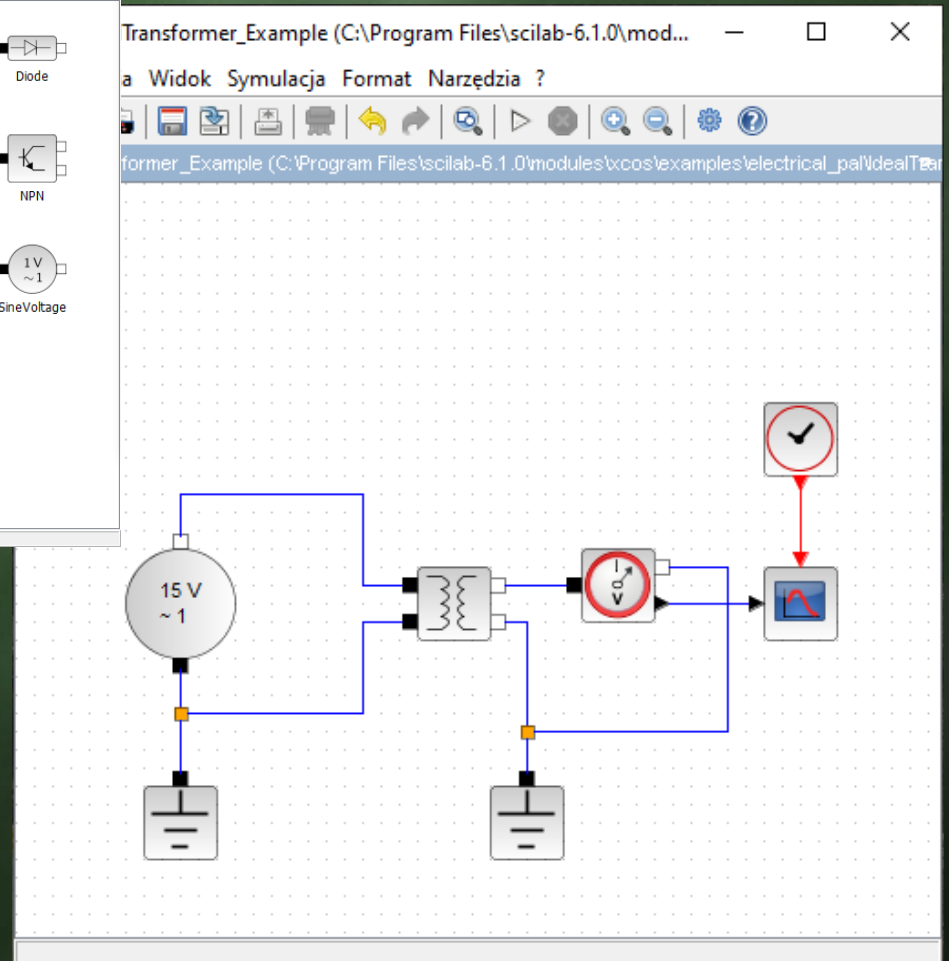
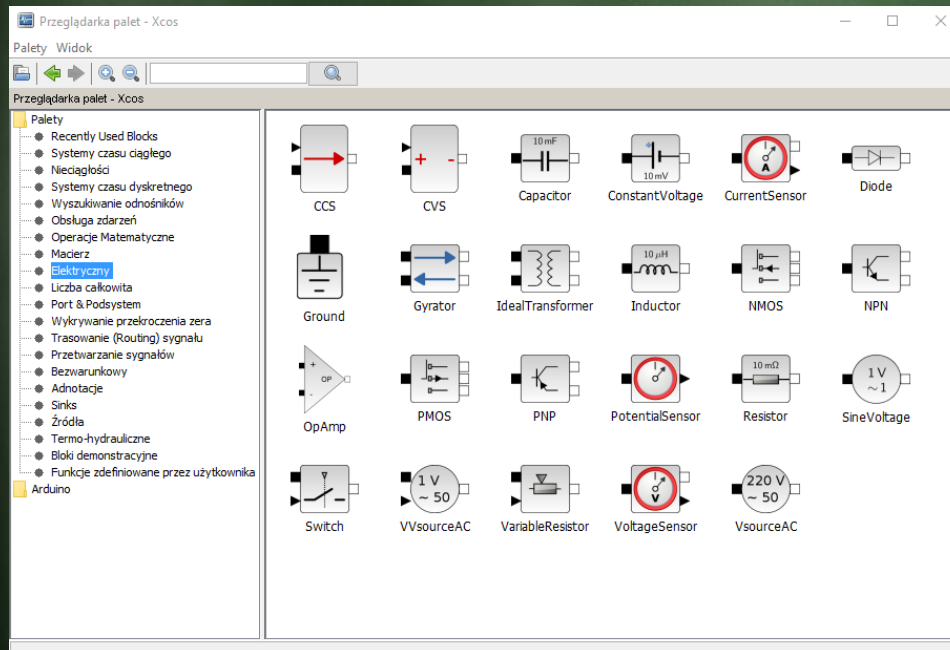


The screenshot shows the Scilab Scinotes editor window. The title bar indicates the file is 'szereg_sin2x_2.sce' located at 'D:\GoogleDrive\Wyklady\Metody numeryczne\Przyklady\sz...'. The menu bar includes 'Plik', 'Edycja', 'Format', 'Ustawienia', 'Okno', and 'Wykonaj ?'. The toolbar contains various icons for file operations, editing, and execution. The main text area displays a MATLAB script with 21 lines of code. The script initializes variables, uses a loop to calculate terms of a series, and plots the result. The code is as follows:

```
1 clear;
2 clc;
3 n=input('Podaj stopien przyblizenia: ');
4 k=1;
5
6 for i=1:1:1000
7     x(i)=0;
8 end
9 for j=1:1:1000
10    t(j)=((%pi)/1000)*(j-1);
11
12    for i=0:1:n
13        w=(((-1)^i)*(2^(2*i+1))*((t(j))^(2*i+1)))/(prod(1:2*i+1));
14        x(j)=x(j)+w;
15    end
16 end
17 plot2d(t,x,);
18 ster_axe=gca();
19 ster_axe.x_location="origin";
20 ster_axe.y_location="origin";
21
```


Scilab - XCos

11:20



Główne kategorie - ATOMS

Plik ?

- Wszystkie moduły
- Bioinformatics
- Contributed Scilab builds
- Data Analysis
- Data Handling
- Documentation
- Domain-Specific
- Education
- GUI
- Graphics
- Graphs
- Image Processing
- Instruments Control
- Linear algebra
- Manuals
- Modeling
- Numerical Maths
- Optimization
- Parallel Computing
- Physics
- Real-Time
- Scilab development
- Przetwarzanie sygnałów
- Windows Tools
- Xcos

Scilab 6.1.0 Console

Plik Edycja Sterowanie Narzędzia ?

Przeglądarka plików

C:\Users\E419_1

Nazwa

- Documents
- ..
- Arduino
- ArduinoData
- Downloads
- Inventor
- MATLAB

SciNotes

Xcos

Translator Matlab do Scilab

ATOMS - menadżer modułów zewn.

Przeglądarka zmiennych

Historia poleceń

Przeglądarka plików

Arduino

Translate blocks interfaces and error messages

- * Translate demos

CHANGES 1.5 => 1.6.2

=====

BUGS FIXED

- * toolbox_arduino_v4.ino file shipped in the module was bad. Replaced with the right file.
- * ARDUINO_pre_simulate() was expecting arduino_v3.ino instead of arduino_v4.ino and yielded an error 'Il faut charger avec le logiciel arduino le sketch toolbox_arduino_v3.ino dans la carte Arduino' when actually dealing with the v.4
- * The DESCRIPTION file that was removed from v.1.4 is actually needed to instal the module offline. Restored.
- * Demos: Examples for Scilab 6.0 are displayed with wrong ports positions with Scilab 5.5. The versions for Arduino 1.1 are restored in the binary for the Scilab 5.5.

TODO:

- * Extend the module to Linux users (through the serial toolbox based on TCL)
- * Translate help pages in english
- * Translate blocks interfaces and error messages
- * Translate demos

Zobacz także

- <http://atoms.scilab.org/toolboxes/arduino/1.8>
- <http://www.scilab-enterprises.com/>

Data wydania

2020-02-16

Wielkość pliku pobranego

1.7 MB

☒ Autoload

Zainstaluj

Usuń

Powrót

Matlab

<https://www.pollub.pl/pl/uczelnia/organizacja-uczelni/jednostki-ogolnouczelniane/ci/matlab>

Does your school have a MATLAB X History | Lublin University of Technology X +

https://www.mathworks.com/academia/tah-support-program/ Szukaj

AstroCenter - AstroPri... Setup

MathWorks® Get MATLAB

Academia

Campus-Wide License

See if your school has a MATLAB campus license

The Campus-Wide License offers an effective way for students, faculty, and researchers to get access to a comprehensive set of MATLAB and Simulink products. To find out if your school is covered under a Campus-Wide License, complete the form below.

* Indicates required information

Information

* University

Lublin University of Technology

Enter the official name.

* Email

Adres e-mail z domeny @pollub.edu.pl

Use your official university email address, which is required for license verification.

Submit

United States

Patents | Trademarks | Privacy Policy | Preventing Piracy | Application Status

© 1994-2020 The MathWorks, Inc.

Join the conversation

Licencja czasowa, odnawiana co trzy lata.

**R2020b**

Advanced Options ▾



LICENSING

DESTINATION

PRODUCTS

OPTIONS

CONFIRMATION

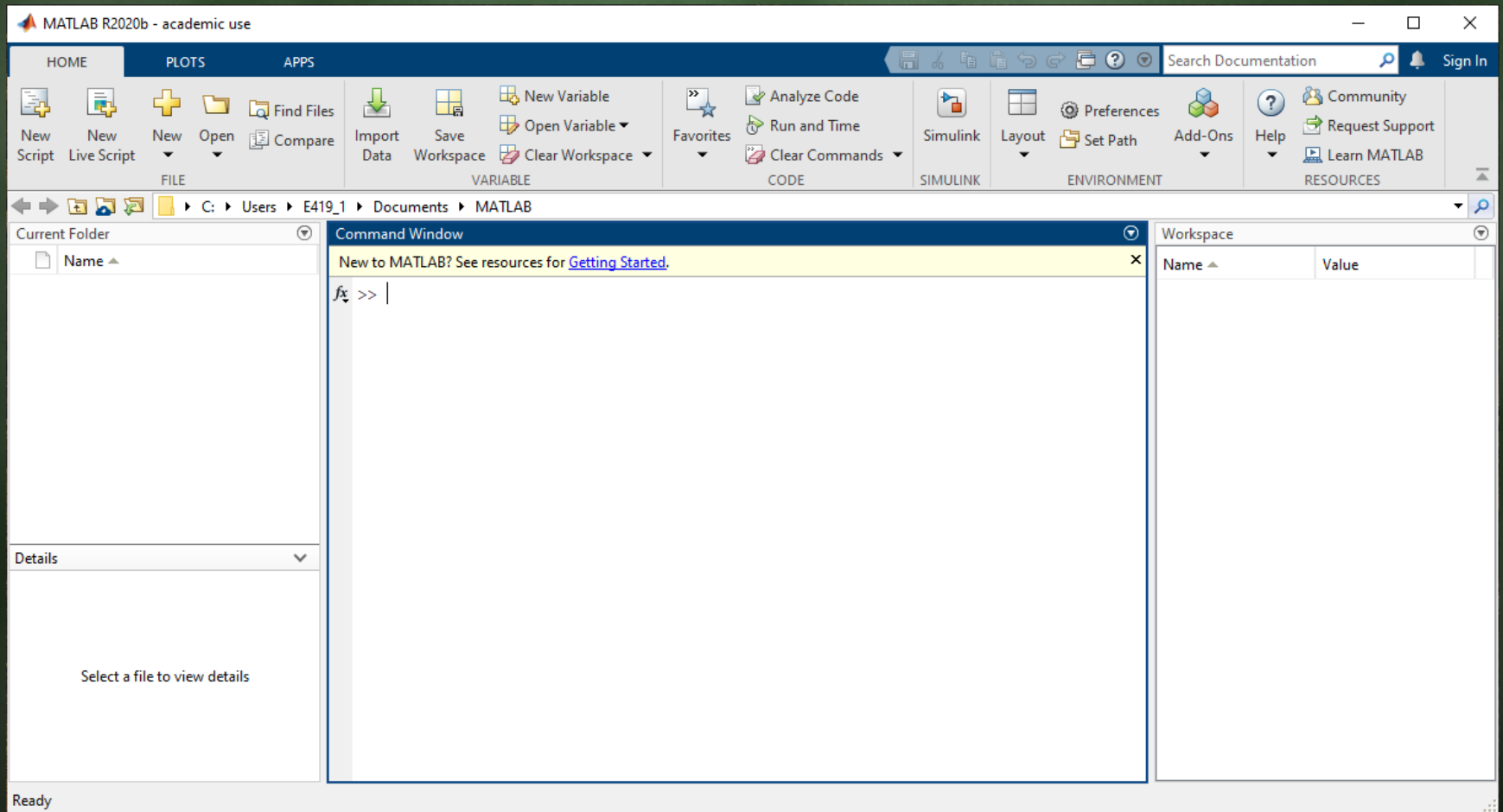
Select products (recommended products are preselected)

☐	Select All
☒	MATLAB
☒	Simulink
☐	5G Toolbox
☐	Aerospace Blockset
☐	Aerospace Toolbox
☐	Antenna Toolbox
☐	Audio Toolbox
☐	Automated Driving Toolbox
☐	AUTOSAR Blockset
☐	Bioinformatics Toolbox
☐	Communications Toolbox

Next

Matlab

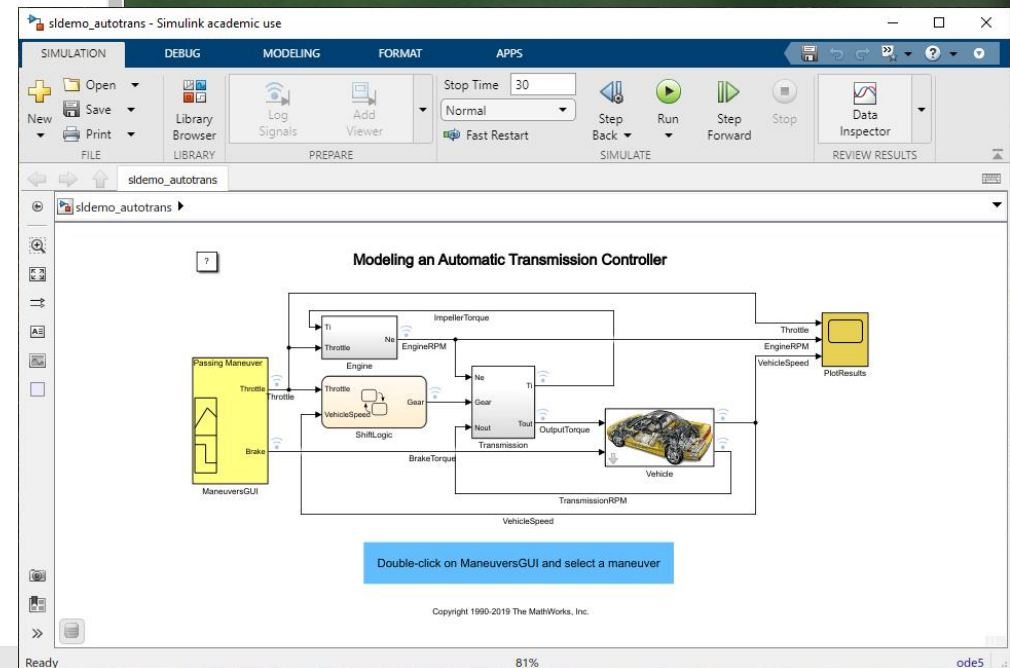
11:20



Matlab - Simulink

The Simulink Start Page displays a sidebar with navigation options: Open..., Recent (sldemo_autotrans.slx), Projects (From Source Control), and Learn (Simulink Onramp, Stateflow Onramp). The main area features a search bar and a grid of example models under the 'Simulink' category. The models include:

- Modeling an Automatic Transmiss...
- Simulation of a Bouncing Ball
- Modeling a Fault-Tolerant Fuel Co... (with an 'Open Example' button)
- Aircraft Longitudinal Flight Control
- Anti-Windup Control Using a PID ...
- Single Hydraulic Cylinder Simulati...
- Radar Tracking Using MATLAB Fu...
- Approximating Nonlinear Relation...
- Simulating Automatic Climate Con...
- Modeling an Anti-Lock Braking Sy...



Metody numeryczne

- Metody rozwiązywania problemów matematycznych za pomocą operacji na liczbach.
- Otrzymywane tą drogą wyniki są na ogół przybliżone, jednak dokładność obliczeń może być **z góry określona i dobiera się ją zależnie od potrzeb.**
- Metody numeryczne wykorzystywane są wówczas, gdy badany problem **nie ma w ogóle rozwiązania analitycznego** (danego równaniem) lub korzystanie z takich rozwiązań jest **uciążliwe** ze względu na ich złożoność.

Zastosowanie

- całkowania
- znajdowania miejsc zerowych wielomianów stopnia większego niż 2 (**korzystanie ze wzorów na dokładne wartości pierwiastków równań stopnia 3 i stopnia 4 jest niepraktyczne, dla równań stopnia wyższego niż 4 wzorów już nie ma**)
- rozwiązywania układów równań liniowych dla większej liczby równań i niewiadomych
- rozwiązywania równań różniczkowych i układów takich równań
- znajdowania wartości i wektorów własnych
- aproksymacji i interpolacja

Znaczenie analizy numerycznej

Właściwie dobrane i zastosowane metody numeryczne umożliwiają symulację zjawisk rzeczywistych.

Katastrofa rakiety Ariane 5
500 milionów \$ (4.06.1996)

Błędna konwersja 64-bit
liczby zmiennoprzecinkowej
na 16-bit liczbę całkowitą

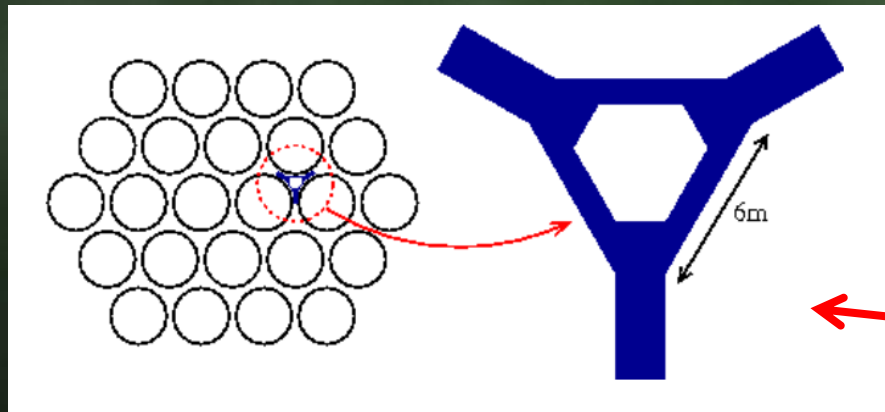


Znaczenie analizy numerycznej

Zatonięcie platformy wiertniczej Sleipner A na Morzu Północnym (23.08.1991)

Koszt – 1 miliard \$

Przyczyna: Niedokładność zamodelowania elementu konstrukcji za pomocą metody elementów skończonych.



Znaczenie analizy numerycznej

Tragedia w Dhahran w Arabii
Saudyjskiej

21.02.1991 – zginęło 28 osób

Błąd w pomiarze czasu (w momencie
wyststrzału wynosił $1/3$ s) spowodował
błąd w pozycjonowaniu celu o 687 m.

Gotowa poprawka systemu dotarła dzień
po tragedii.



Znaczenie analizy numerycznej

NASA Mars Climate Orbiter

23.09.1999 uległ zniszczeniu w atmosferze Marsa MCO, 700 milionów \$

Przyczyna: błąd w oprogramowaniu komunikacyjnym między napędami MCU spowodowany stosowaniem innym jednostek metrycznych przez NASA i firmę brytyjską która napisała oprogramowanie.



Pojęcia podstawowe

Analiza numeryczna – tworzenie, badanie i analiza algorytmów, w celu otrzymania rozwiązania numerycznego różnorodnych zadań matematycznych

Metody numeryczne – tworzenie i badanie algorytmów (procedur) obliczeniowych umożliwiających przeprowadzenie analizy numerycznej

Pojęcia podstawowe

Rozwiązanie analityczne danego zadania może być odmienne od jego rozwiązania numerycznego.

Rozwiązanie analityczne przy adaptacji bezpośredniej do algorytmu numerycznego może być wolno zbieżne, długotrwałe w obliczeniach, a więc kłopotliwe lub bezużyteczne w analizie numerycznej.

Pojęcia podstawowe

Metody numeryczne stosuje się gdy stawiany problem nie posiada rozwiązania analitycznego (określonego wzorem) lub stosowanie takich wzorów jest kłopotliwe ze względu na ich złożoność.

Pojęcia podstawowe

Obliczenia numeryczne

Obliczenia wykonywane są bezpośrednio na liczbach

Obliczenia symboliczne

Przekształcenia zgodne z regułami matematycznymi wykonywane są na symbolach reprezentujących liczby

$$\frac{(35.3)^2 - 1}{35.3 + 1} = 34.3$$

$$\frac{x^2 - 1}{x + 1} = x - 1$$

Pojęcia podstawowe

Rozwiązanie analityczne

Dokładny wynik numeryczny lub symboliczny
(z wykorzystaniem symboli i funkcji matematycznych)

Rozwiązanie numeryczne

Wynik w postaci numerycznej, z określoną
dokładnością

$$\frac{1}{4} \longrightarrow 0.25 \qquad \pi \longrightarrow 3.14159 \dots$$

$$\frac{2}{3} \longrightarrow 0.66667 \dots \quad \text{tg}(36) \longrightarrow 7.75047 \dots$$

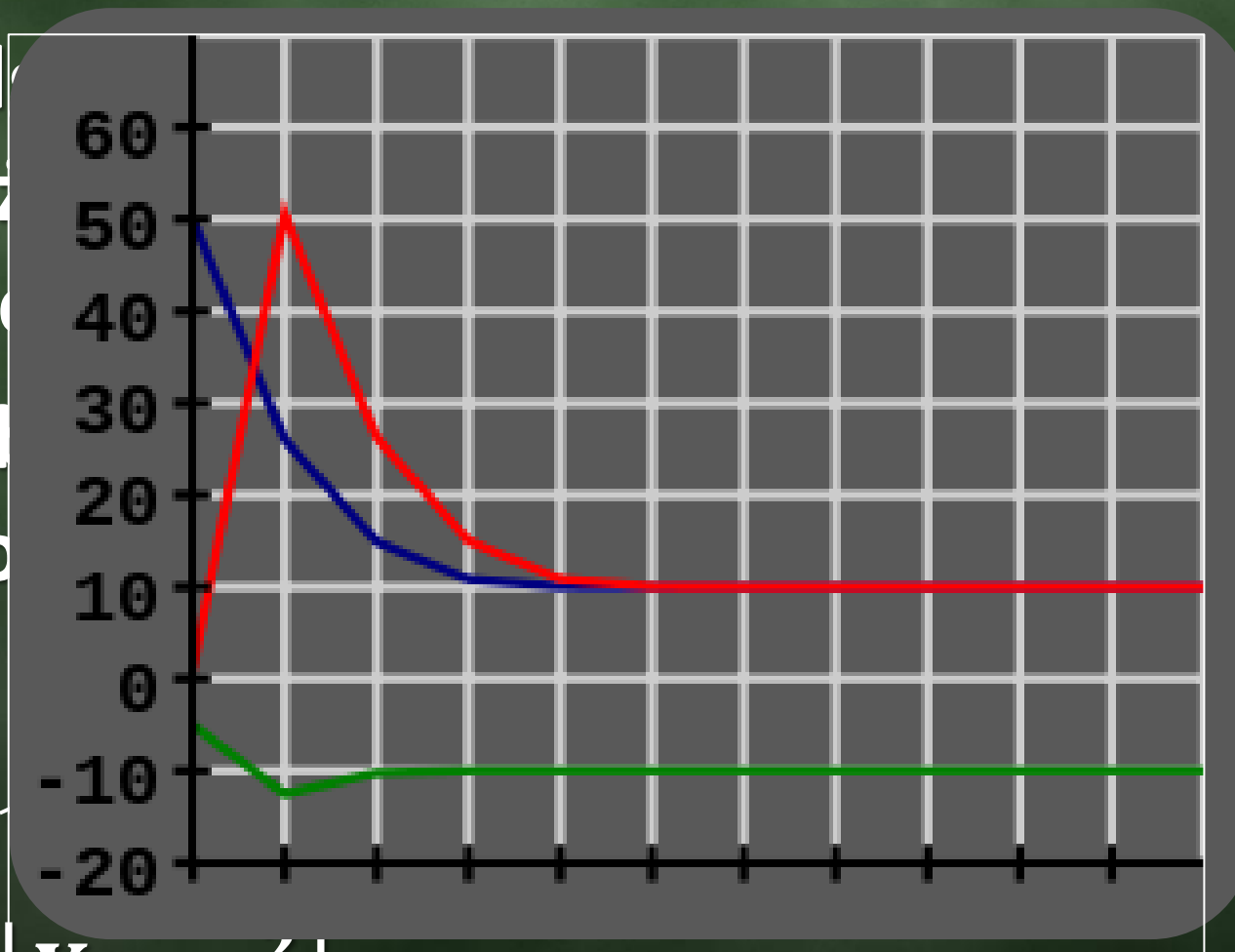
Obliczenia

Al
umoz
warto
Of
meto

yczaj
wanej
wego

$$a =$$

$$\delta \leq |X - a_n^L|$$



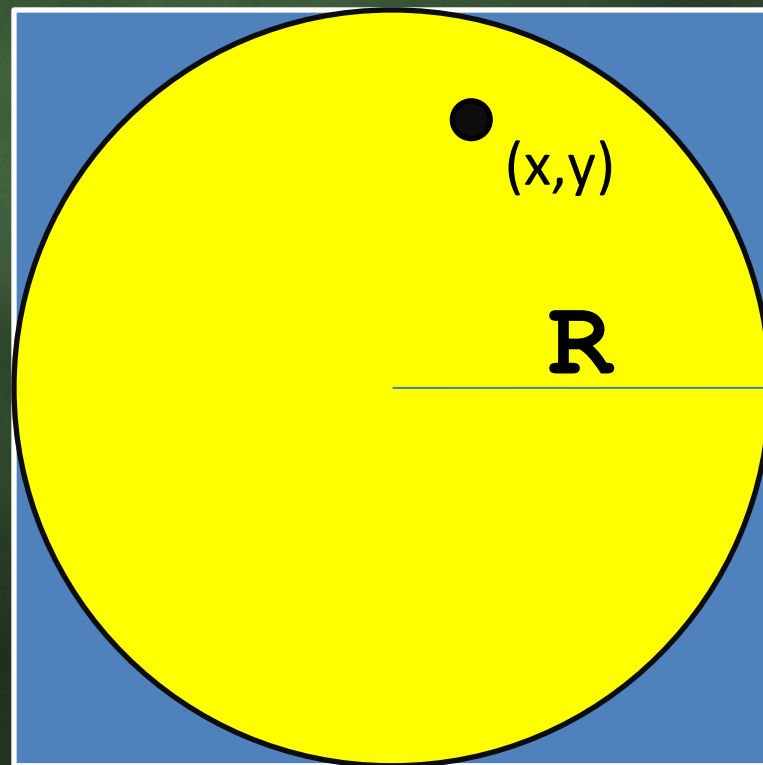
$$\left(\frac{1}{-1} \right)$$

Wyznaczanie liczby π

$$\frac{N_o}{N_{\blacksquare}} \approx \frac{\pi R^2}{4R^2}$$

$$\pi \approx 4 \frac{N_o}{N_{\blacksquare}}$$

$$(x^2 + y^2) \leq R^2$$



Pojęcia podstawowe

Wzór Taylora

Jeżeli funkcja $f(x)$ ma w przedziale $[a,b]$ pochodną rzędu $n-1$ oraz pochodną rzędu n w przedziale (a,b) wówczas istnieje $c \in (a,b)$, takie że

$$f(x) = \sum_{k=0}^n \frac{1}{k!} f^{(k)}(c) (x - c)^k + E_n(x)$$

Pojęcia podstawowe

Dla pewnego punktu α zawartego w przedziale (c, x) można określić :

$$E_n(x) = \frac{(x - c)^{n+1}}{(n + 1)!} f^{(n+1)}(\alpha)$$

zwaną resztą **Lagrange'a** wzoru **Taylora**

Definiuje ona dokładność otrzymanego przybliżenia

Pojęcia podstawowe

Szczególny przypadek wzoru Taylora dla $c=0$ określany jest wzorem Maclaurina

$$f(x) = \sum_{k=0}^n \frac{1}{k!} f^{(k)}(0) x^k + E_n(x)$$

$$E_n(x) = \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\alpha)$$

Pojęcia podstawowe

Pochodne

$$(a)' = 0 \quad (x^n)' = nx^{n-1} \quad (e^x)' = e^x$$

$$(\ln(x))' = \frac{1}{x} \quad (\log_a(x))' = \frac{1}{x \ln(a)}$$

$$(a^x)' = a^x \ln(a), a > 0 \quad (\sin(x))' = \cos(x)$$

$$\cos(x)' = -\sin(x) \quad (\operatorname{tg}(x))' = \frac{1}{\cos^2(x)}$$

$$(f(g(x)))' = f'(g(x))g'(x)$$

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

Pojęcia podstawowe

Szereg Taylora

Na podstawie wzoru Taylora
wyprowadza się szereg wielomianowy
Taylora:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x - x_0)^k}{k!} f^{(k)}(x_0)$$

$$= f(x_0) + \frac{(x - x_0)}{1!} f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \dots$$

Pojęcia podstawowe

Szereg Taylora

1. im wyższy jest jego stopień, tym lepiej (zwykle) aproksymuje funkcję,
2. twierdzenie Taylora, daje informację o błędzie aproksymacji (reszta),
3. wielomiany to funkcje elementarne, które bardzo łatwo wyznaczać numerycznie - z tego względu szereg kalkulatorów (czy programów / języków programowania) implementuje funkcje takie jak $\sin(x)$, $\cos(x)$, e^x , itp. poprzez odpowiednie wielomiany Taylora,

Pojęcia podstawowe

Szereg Taylora

4. wielomiany z łatwością poddają się różniczkowaniu i całkowaniu,
5. wielomiany można zapisać w postaci iloczynowej, ułatwiając rozwiązywanie szeregu równań i nierówności,
6. wielomiany są określone na całej prostej rzeczywistej (zespólonej), co umożliwia analityczne "przedłużanie" wartości funkcji na dziedzinę, w której wyjściowa funkcja jest nieokreślona.

Pojęcia podstawowe

Szereg Maclaurina

Dla $x_0=0$ szereg Taylora przyjmuje postać:

$$f(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} f^{(k)}(0)$$

$$= f(0) + \frac{x}{1!} f'(0) + \frac{(x)^2}{2!} f''(0) + \dots$$

Przykład 1

Rozwinięcie w szereg Taylora funkcji

$$f(x) = \sin(2x)$$

$$f(0) = \sin(0) = 0$$

$$f'''(x) = -8\sin(2x)$$

$$f'(x) = 2\cos(2x)$$

$$f^{(4)}(x) = 16\cos(2x)$$

$$f''(x) = -4\cos(2x)$$

$$f^{(5)}(x) = 32\sin(2x)$$

$$f^{(n)}(x) = \begin{cases} (-1)^k \cdot 2^n \cdot \sin 2x & \text{dla } n = 2k \\ (-1)^k \cdot 2^n \cdot \cos 2x & \text{dla } n = 2k + 1 \end{cases}$$

Przykład 1

Rozwinięcie w szereg Taylora funkcji

$f(x)=\sin(2x)$ dla $x=0$

$$f(0)=\sin(0)=0$$

$$f'(x)=2$$

$$f''(x)=0$$

$$f'''(x)=-8$$

$$f^{(4)}(x)=0$$

$$f^{(5)}(x)=32$$

$$f^{(n)}(0) = \begin{cases} 0 & \text{dla } n = 2k \\ (-1)^k \cdot 2^n & \text{dla } n = 2k + 1 \end{cases}$$

$$(k = 0, 1, 2, \dots)$$

Przykład 1

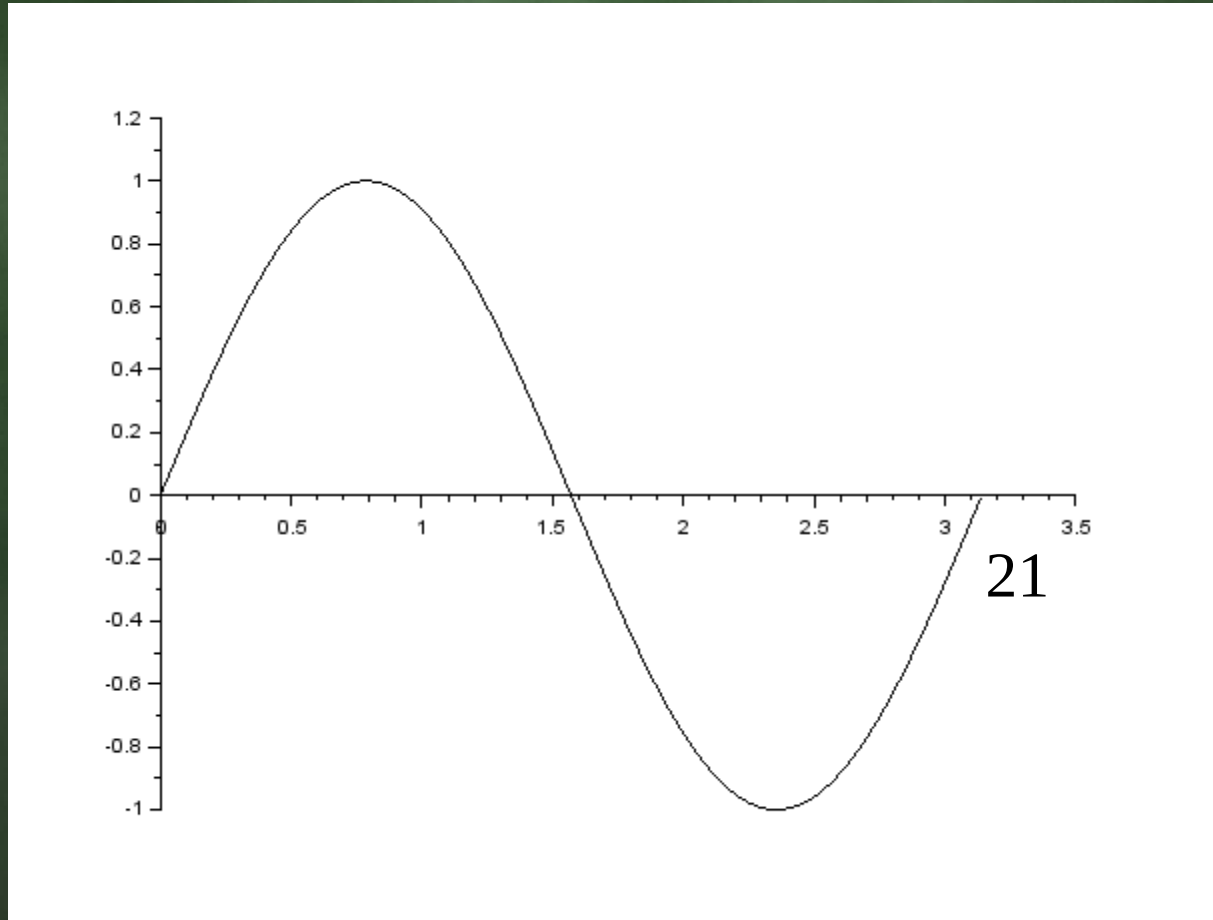
Rozwinięcie w szereg Maclaurina

$$\sum_{n=0}^{\infty} \frac{(-1)^k 2^{2k+1} x^{2k+1}}{(2k+1)!}$$

n – przyjmuje tylko wartości z szeregu
2k+1

(1, 3, 5, 7, ..., 2k+1)

Przykład 1



Stopnie przybliżenia:

$$n=2k+1=1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21$$

Pojęcia podstawowe

Rozwinięcia w szereg Taylora podstawowych funkcji matematycznych

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \qquad \cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

$$\sqrt{x+1} = \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{(1-2n)(n!)^2 4^n} x^n, |x| < 1$$

$$\arcsin(x) = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1}, |x| \leq 1$$

Przykład 2

Za pomocą wzoru Taylora można wyznaczać przybliżone wartości funkcji.

$$\sqrt{x} = \sum_{k=0}^n \frac{(x - x_0)^k}{k!} f^{(k)}(x_0)$$

$\sqrt{10}$, można wyznaczyć dla $x_0=9$

$$\sqrt{10} = \sum_{k=0}^n \frac{(10 - 9)^k}{k!} (\sqrt{9})^{(k)} = \sum_{k=0}^n \frac{(\sqrt{9})^{(k)}}{k!}$$

Przykład 2

Kolejne iteracje rozwinięcia:

$$k = 0 \rightarrow \sqrt{9} \qquad k = 1 \rightarrow \frac{(\sqrt{9})^{(1)}}{1!} = \frac{1}{2\sqrt{9}}$$

$$k = 2 \rightarrow \frac{(\sqrt{9})^{(2)}}{2!} = -\frac{1}{8\sqrt{9}^3}$$

$$k = 3 \rightarrow \frac{(\sqrt{9})^{(3)}}{3!} = \frac{3}{48\sqrt{9}^5}$$

$$k = 4 \rightarrow \frac{(\sqrt{9})^{(4)}}{4!} = -\frac{15}{384\sqrt{9}^7}$$

Przykład 2

Wynika ostateczny

$$\sqrt{10} = \sqrt{9} + \frac{1}{6} - \frac{1}{216} + \frac{3}{11664} - \frac{15}{7558272}$$

$$\sqrt{10} = 3 + \frac{1}{6} - \frac{1}{216} + \frac{1}{3888} - \frac{5}{2519424}$$

$$\sqrt{10} = \underline{3,162292254}$$

Pojęcia podstawowe

Inny warianty wzoru Taylora można osiągnąć zamieniając **x** na **x+h** i **c** na **x**

$$f(x + h) = \sum_{k=0}^n \frac{h^k}{k!} f^k(x) + E_n(h)$$

$$E_n(h) = \frac{h^{n+1}}{(n+1)!} f^{n+1}(\alpha)$$

Punkt **α** leży pomiędzy **x** i **x+h**

Przykład 3

W oparciu o przedstawiony wariant wzoru Taylora można obliczyć przybliżoną wartość funkcji dla wartości rzeczywistej.

Obliczyć $\sqrt{134.65}$

Najbliższym znanym całkowitym rozwiązaniem pierwiastka dającym wynik całkowity jest $\sqrt{144} = 12$

Przykład 3

$$\sqrt{144 - 9.35} = \sum_{k=0}^n \frac{(-9.35)^k}{k!} (\sqrt{144})^{(k)}$$

$$12 + \frac{(-9.35)}{2\sqrt{144}} - \frac{(-9.35)^2}{2! 4(\sqrt{144})^3} + \frac{3(-9.35)^3}{3! 8(\sqrt{144})^5}$$

$$12 - 0.38958 - 0.006324 - 0.00021$$

$$\sqrt{144 - 9.35} = 11.60389$$

$$\sqrt{134.65} = 11.60388$$

Wzór Taylora

1. Im wyższy stopień wielomianu tym dokładniejsza aproksymacja
2. Rozwinięcie funkcji w wielomianowy szereg Taylora ułatwia jej całkowanie i różniczkowanie.
3. Rozwinięcie w szereg Taylora daje informacje o błędzie aproksymacji
4. Umożliwia numeryczne obliczenia z funkcjami elementarnymi e^x , $\sin(x)$, $\cos(x)$ itp.
5. Wielomiany można zapisać w postaci iloczynowej, ułatwiając rozwiązywanie szeregu równań i nierówności.
6. Wielomiany są określone na całej prostej rzeczywistej (lub płaszczyźnie zespolonej), co umożliwia analityczne „przedłużanie” wartości funkcji na dziedzinę, w której wyjściowa funkcja jest nieokreślona.

Pojęcia podstawowe

W obliczeniach numerycznych często zamiast ostatecznego wyniku otrzymuje się ciąg przybliżonych rozwiązań, zazwyczaj coraz dokładniejszych.

Rozwiązanie określamy wtedy jako:

$$\lim_{n \rightarrow \infty} x_n = L,$$

a wynik określamy w sytuacji gdy :

$$|x_n - L| < \varepsilon$$

Pojęcia podstawowe

Rozwiązania obliczeń numerycznych posiadają zazwyczaj więc rozwiązanie w postaci ciągu zbieżnego.

Poszukiwane rozwiązanie może być zbieżne:

1. liniowo
2. nadliniowo
3. kwadratowa
4. rzędu α

Pojęcia podstawowe

Zbieżność liniowa

Jeżeli istnieje stała $c < 1$ i liczba całkowita N takie że:

$$|x_{n+1} - x^*| \leq c |x_n - x^*| \quad (n \geq N)$$

gdzie, x^* jest wartością (rzeczywistą) do jakiej zbieżny jest ciąg rozwiązań.

Pojęcia podstawowe

Zbieżność nadliniowa

Jeżeli istnieje ciąg ε_n zbieżny do zera i liczba całkowita N takie że:

$$|x_{n+1} - x^*| \leq \varepsilon_n |x_n - x^*| \quad (n \geq N)$$

gdzie, x^* jest wartością (rzeczywistą) do jakiej zbieżny jest ciąg rozwiązań.

Pojęcia podstawowe

Zbieżność kwadratowa

Jeżeli istnieją stała $C > 0$ i liczba całkowita N takie że:

$$|x_{n+1} - x^*| \leq C |x_n - x^*|^2 \quad (n \geq N)$$

gdzie, x^* jest wartością (rzeczywistą) do jakiej zbieżny jest ciąg rozwiązań.

Pojęcia podstawowe

Zbieżność rzędu α

Jeżeli istnieją stała $C > 0$, stała $\alpha > 1$ oraz liczba całkowita N takie że:

$$|x_{n+1} - x^*| \leq C |x_n - x^*|^\alpha \quad (n \geq N)$$

gdzie, x^* jest wartością (rzeczywistą) do jakiej zbieżny jest ciąg rozwiązań.

Dla α wynoszących:

$1 < \alpha < 2$ – superliniowa

$\alpha = 3$ – kubiczna

Pojęcia podstawowe

Równania różnicowe

Pod pojęciem równania różnicowego rozumiemy związek pomiędzy kilkoma elementami ciągu rekurencyjnego, natomiast jego rozwiązaniem jest **n**-ty wyraz tego ciągu.

Ciągiem rekurencyjnym określamy taki ciąg którego każdy kolejny wyraz zdefiniowany jest poprzez odwołanie do poprzednich .

Pojęcia podstawowe

Równania różnicowe

Symbol **iloczynu** pewnej liczby
wyrazów w ciągu:

$$\prod_{k=1}^n a(k) = a(1) \cdot a(2) \cdot a(3) \cdot a(4) \cdot a(5) \cdot \dots a(n)$$

Operator przesunięcia **E** określony na
ciągach:

$$Ex(n) = x(n+1) \quad E^k x(n) = x(n+k), k \in \mathbb{N}$$

$$E^2 x(n) = E(Ex(n)) = Ex(n+1) = x(n+2) \quad 60$$

Pojęcia podstawowe

Równania różnicowe

Operator identyczności /

$$Ix(n) = x(n)$$

Postać ogólna wielomianu stopnia **k**
zmiennnej **λ**

$$p(\lambda) = a_0\lambda^k + a_1\lambda^{k-1} + a_2\lambda^{k-2} + \dots + a_k$$

Operator wielomianowy **$p(E)$**
przyjmuje postać:

$$p(E) = a_0E^k + a_1E^{k-1} + a_2E^{k-2} + \dots + a_kI$$

Pojęcia podstawowe

Równania różnicowe

Na ciągu $x(n)$ operator wielomianowy przyjmuje postać:

$$p(E)x(n) =$$

$$a_0x(n+k) + a_1x(n+k-1) + \cdots + a_kx(n)$$

Przykład 4

W chwili $t=0$ pewna populacja liczy $P(0)$. Roczny wskaźnik urodzeń wynosi b , a współczynnik umieralności d . Zapisać równanie różnicowe opisujące przyrost populacji.

Jeżeli dla n -tego roku stan populacji wynosi $P(n)$ to:

$$\begin{aligned} P(n+1) &= P(n) + b \cdot P(n) - d \cdot P(n) = \\ &= P(n)(1 + b - d) \end{aligned}$$

Przykład 4

Dążąc do postaci ogólnej ciągu $P(n)$
wprowadzamy oznaczenie $r=b-d$

$$P(n+1) = P(n)(1+r)$$

$$P(n+2) = P(n+1)(1+r) =$$

$$P(n)(1+r)(1+r) = P(n)(1+r)^2$$

Można więc zaproponować postać
ogólną równania różnicowego:

$$P(n) = A(1+r)^n, n = 0, 1, 2, \dots$$

Przykład 4

Na podstawie warunku początkowego dla $n=0$, $P(n)=P(0)$ można wyznaczyć wartość stałej A :

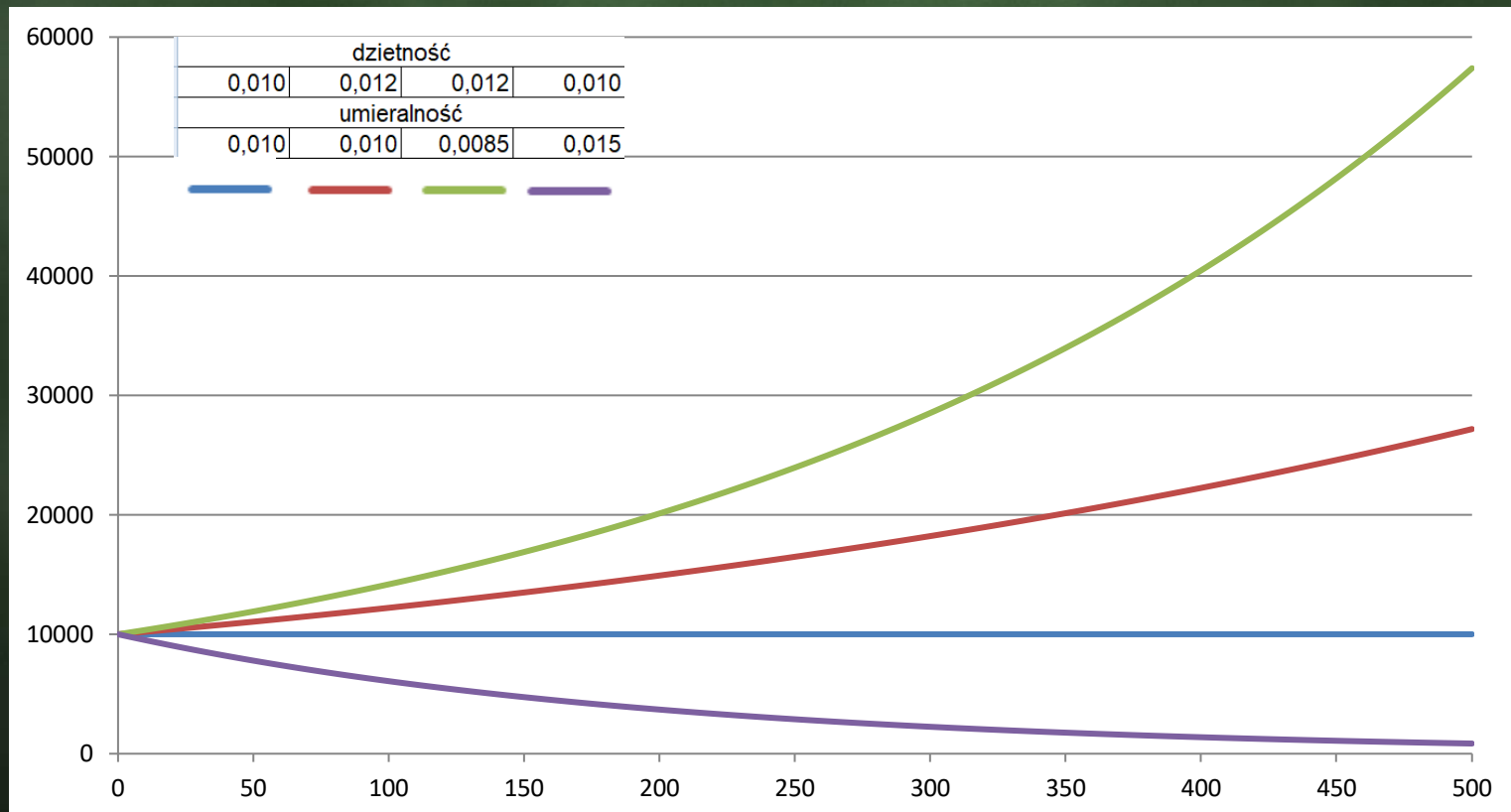
$$P(0) = A(1 + r)^0 = A$$

Równanie różnicowe dla postawionego problemu przyjmie więc postać:

$$P(n) = P(0)(1 + r)^n$$

Przykład 4

Dla $P(0)=10\ 000$ oraz dzietnością i umieralnością opisanymi w tabeli można wykreślić rodzinę krzywych wzrostu populacji.



Arytmetyka zmiennopozycyjna

- Systemy komputerowe pracują w oparciu o system dwójkowy.
- Ze względu na to że w powszechnym użyciu stosujemy system dziesiętny w systemach komputerowych następuje konwersja pomiędzy tymi systemami,
 - na wejściu
 - na wyjściu danych.
- Z tego względu mogą pojawić się błędy z tym związane.

Arytmetyka zmiennopozycyjna

W systemie dziesiętnym każda liczba rzeczywista zapisywana jest w postaci ciągu, wraz z odpowiednim znakiem (+/-):

$$23.679 = 2 \cdot 10^1 + 3 \cdot 10^0 + 6 \cdot 10^{-1} + 7 \cdot 10^{-2} + 9 \cdot 10^{-3}$$

W układzie dwójkowym stosuje się tylko cyfry 0 i 1

$$\begin{aligned}(1001.11101)_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + \\ &1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} = \\ &= (9.90625)_{10}\end{aligned}$$

Arytmetyka zmiennopozycyjna

Pozostałe stosowane systemy liczbowe:

1. Ósemkowy – N_8 – cyfry 0 – 7
2. Szesnastkowym – N_{16} – cyfry 0 – 9 i litery A – F

Można w zależności od potrzeb zbudować dowolny system liczbowy którego postać ogólna N_β w oparciu o zakres liczbowy 0 – $(\beta-1)$ i podstawie $\beta > 1$

Arytmetyka zmiennopozycyjna

Systemy komputerowe z racji swoich ograniczeń nie potrafią operować na nieskończonych rozwinięciach liczb w systemie binarnym.

$$\frac{1}{10} = (0.0\ 0011\ 0011\ 0011\ 0011\ \dots)_2$$

Przy konwersji na system 10 z dokładnością do 40 cyfr dziesiętnych otrzymamy:

0.10000 00014 90116 11938 47656 25000 00000 00000

Zaokrąglenia

Zaokrąglanie jest istotnym zagadnieniem obliczeń naukowych

Podstawowy system zaokrąglania do n -tej cyfry:

- Jeżeli cyfra $n+1$ ma wartość 0, 1, 2, 3, 4 to n cyfra pozostaje bez zmiany
- Jeżeli cyfra $n+1$ ma wartość 5, 6, 7, 8, 9 to wartość n -tej cyfry zostaje zwiększona o 1
- Jeżeli cyfra $n+1$ ma wartość 5, to czasami zwiększenie o 1 realizuje się tylko w sytuacji gdy zwiększenie n -tej liczby da w wyniku liczbę parzystą

Zaokrąglenia

Zaokrąglenie liczby siedmiocyfrowej
do liczby czterocyfrowej:

0.1783**4**98 $\longrightarrow$ 0.1783

0.9999**5**00 $\longrightarrow$ 1.0000

0.4321**4**09 $\longrightarrow$ 0.4321

0.5767**5**23 $\longrightarrow$ 0.5768

0.5768**5**23 $\longrightarrow$ 0.5768

Liczby zmiennopozycyjne

W systemie **dziesiętnym** każdą liczbę rzeczywistą można przedstawić w postaci zmiennopozycyjnej (zmiennoprzecinkowej):

$$x = \pm r \times 10^n$$

gdzie $r \in [1,10]$ (lub czasami $r \in [0.1,1]$), a n jest liczbą całkowitą $+$, $-$ lub 0 .

Jeżeli $x=0$ to $r=0$, a n przyjmuje wartość dowolną.

Liczby zmiennopozycyjne

W systemie **binarnym** zmiennopozycyjna postać liczby ($x \neq 0$) zapisana jest analogicznie jak w systemie dziesiętnym:

$$x = (-1)^z m \times 2^c$$

gdzie $m \in [1, 2]$ (lub czasami $q \in [0.5, 1]$), a c jest liczbą całkowitą. Przyjmuje się, że mantysa m i cecha c są wyrażone w układzie dwójkowym, z jest bitem znaku.

Liczby w systemach komputerowych opisane są jak wyżej, z ograniczeniami wartości m i c wynikającymi z długości słowa

Liczby zmiennopozycyjne

Zgodnie ze standardem IEEE (Nr. 754 z 1985 r.)
liczba zmiennopozycyjna binarna składa się:

Format zapisu zmiennopozycyjnego IEEE 754			
32 bity Pojedyncza precyzja	(1 bit) b_{31}	(8 bitów) (BIAS=127) $b_{30} \dots b_{23}$	(23 bitów) $b_{22} \dots b_0$
64 bity Podwójna precyzja	(1 bit) b_{63}	(11 bitów) (BIAS=1023) $b_{62} \dots b_{52}$	(52 bitów) $b_{51} \dots b_0$
Opis pól bitowych	Bit znaku z	Bity kodu cechy c	Bity ułamkowe mantysy m

Liczby w tej postaci określa się często jako
liczby maszynowe

Liczby zmiennopozycyjne

Dla systemu 32 bitowego możliwe jest zapamiętanie liczb z przedziału:

$$2^{-126} \rightarrow (2 - 2^{-23})2^{127}$$

$$1.2 \times 10^{-38} \rightarrow 3.4 \times 10^{38}$$

Dla liczb dodatnich i ujemnych

Liczby zmiennopozycyjne

Przyjmuje się, że cecha **c** może przyjmować wartości całkowite z zakresu $[-126, 127]$, stąd $(c+127) \in [1, 254]$. Na 8 bitach można zapisać liczby z zakresu $[0, 255]$.

$$0 \rightarrow (00\ 000\ 000)_2$$

$$255 \rightarrow (11\ 111\ 111)_2$$

Liczby zmiennopozycyjne

Mantysa liczby różnej od zera ma z definicji część całkowitą równą 1, z tego względu zapamiętuje się tylko część ułamkową:

1 → 00 000 000 000 000 000 000 000 000

$2 \cdot 2^{-23}$ → 11 111 111 111 111 111 111 111

Przykład 1

Przeliczanie liczby 125.5 z systemu dziesiętnego na binarny zmiennopozycyjny.

1. Wyznaczenie cechy **c** i mantysy **m** (dziesiętny)
2. Konwersja cechy **c** na system binarny
3. Konwersja mantysy **m** w na system binarny
4. Zapisanie liczby w postaci binarnej zmiennopozycyjnej

Przykład 1

Wyznaczanie cechy i mantysy

$$125.5 \times 2^0$$

$$62.75 \times 2^1$$

$$31.375 \times 2^2$$

$$15.6875 \times 2^3$$

$$7.84375 \times 2^4$$

$$1.9609375 \times 2^5$$

Mantysa $m = 0.9609375$

Cecha $c = 5 + 127 = 132$

Bit znaku $z = 0$

Przykład 1

Konwersja mantysy realizowana jest poprzez mnożenie części ułamkowej liczby przez 2. Jeżeli wynik jest większy od zera składnik mantysy ustawiamy na 1, jeżeli mniejszy od jedynki ustawiamy 0.

Przykład 1

Określanie binarnej postaci mantysy

1. $0.9609375 \times 2 = 1,921875 \rightarrow 1$
2. $0.921875 \times 2 = 1,84375 \rightarrow 1$
3. $0.84375 \times 2 = 1,6875 \rightarrow 1$
4. $0.6875 \times 2 = 1,375 \rightarrow 1$
5. $0.375 \times 2 = 0.75 \rightarrow 0$
6. $0.75 \times 2 = 1,5 \rightarrow 1$
7. $0.5 \times 2 = 1,0 \rightarrow 1$
8. $0 \times 2 = 0 \rightarrow 0$

Przykład 1

Określanie binarnej postaci cechy

$$132 \text{ div } 2 = 66; \quad 132 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$66 \text{ div } 2 = 33; \quad 66 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$33 \text{ div } 2 = 16; \quad 33 \text{ mod } 2 = 1 \quad \rightarrow 1$$

$$16 \text{ div } 2 = 8; \quad 16 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$8 \text{ div } 2 = 4; \quad 8 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$4 \text{ div } 2 = 2; \quad 4 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$2 \text{ div } 2 = 1; \quad 2 \text{ mod } 2 = 0 \quad \rightarrow 0$$

$$1 \text{ div } 2 = 0; \quad 1 \text{ mod } 2 = 1 \quad \rightarrow 1$$

$$c = 10000100$$

Przykład 1

Ostateczna liczba 125.5 ma postać
binarną zmiennopozycyjną 32 bit:

125.5=

0 10000100 111101100000000000000000

Przykład 2

Przeprowadzić konwersję liczby binarnej zmiennopozycyjnej na postać dziesiętną.

1 10001100 100111100100101000000000

z=1

c=10001100

m=100111100100101000000000

Przykład 2

Po identyfikacji elementów, mantysy, cechu i bitu znaku, należy przeprowadzić konwersję z postaci binarnej na dziesiętną mantysy i cechy. Należy pamiętać że mantysa przechowuje informację o części ułamkowej i należy po konwersji dodać do niej 1, natomiast od wyznaczonej wartości cechy należy odjąć 127.

Przykład 2

Przeliczanie cechy $c=10001100$

$$c=2^2+2^3+2^7-127=4+8+128-127=13$$

Przeliczanie mantysy

$$m=10011100100101$$

$$m=1+2^{-1}+2^{-4}+2^{-5}+2^{-6}+2^{-7}+2^{-10}+2^{-13}+2^{-15}$$

$$=1,618316650390625$$

Przykład 3

Liczba binarna zmiennopozycyjna:

1 10001100 100111100100101000000000

po konwersji wynosi:

$$(-1)^z \times m \times 2^c =$$

$$=(-1)^1 \times 1,618316650390625 \times 2^{13} =$$

$$=-13257.25$$

Liczby zmiennopozycyjne

Nie każdą liczbę da się przedstawić w postaci zmiennopozycyjnej bitowej o skończonym rozwinięciu jak w przykładach wcześniejszych. W efekcie nie da się zapisać liczby w sposób dokładny. Powoduje to że zapis na możliwie bliska liczbę maszynową jest obarczony błędem.

W przypadku postaci 32 bitowej mamy 23 bity mantysy, natomiast 64 bitowej liczba bitów mantysy wynosi 52.

Przykład 3

Przeprowadzić konwersję liczby **0.01** do postaci zmiennopozycyjnej binarnej.

Bit znaku **$z=0$** **$c=-7+127=120$** **$m=0.28$**

$$0,01 * 2^0 = 0,01$$

$$0,02 * 2^{-1} = 0,01$$

$$0,04 * 2^{-2} = 0,01$$

$$0,08 * 2^{-3} = 0,01$$

$$0,16 * 2^{-4} = 0,01$$

$$0,32 * 2^{-5} = 0,01$$

$$0,64 * 2^{-6} = 0,01$$

$$1,28 * 2^{-7} = 0,01$$

Przykład 3

11:21

c			
1	120	60	0
2	60	30	0
3	30	15	0
4	15	7,5	1
5	7	3,5	1
6	3	1,5	1
7	1	0,5	1
8	0	0	0

c = 01111000

m			
1	0,28	0,56	0
2	0,56	1,12	1
3	0,12	0,24	0
4	0,24	0,48	0
5	0,48	0,96	0
6	0,96	1,92	1
7	0,92	1,84	1
8	0,84	1,68	1
9	0,68	1,36	1
10	0,36	0,72	0
11	0,72	1,44	1
12	0,44	0,88	0
13	0,88	1,76	1
14	0,76	1,52	1
15	0,52	1,04	1

29	0,68	1,36	1
30	0,36	0,72	0
31	0,72	1,44	1
32	0,4400001	0,88	0
33	0,8800001	1,76	1
34	0,7600002	1,52	1
35	0,5200005	1,040001	1
36	0,0400009	0,080002	0
37	0,0800018	0,160004	0
38	0,1600037	0,320007	0
39	0,3200073	0,640015	0
40	0,6400146	1,280029	1
41	0,2800293	0,560059	0
42	0,5600586	1,120117	1

11:21

m=01000111101011100001010

19	0,32	0,64	0
20	0,64	1,28	1
21	0,28	0,56	0
22	0,56	1,12	1
23	0,12	0,24	0
24	0,24	0,48	0
25	0,48	0,96	0
26	0,96	1,92	1
27	0,92	1,84	1
28	0,84	1,68	1

43	0,4800000	0,9600000	0
46	0,9609375	1,921875	1
47	0,921875	1,84375	1
48	0,84375	1,6875	1
49	0,6875	1,375	1
50	0,375	0,75	0
51	0,75	1,5	1
52	0,5	1	1
53	0	0	0

Przykład 3

0.01=

0 01111000 010001111010111000001010

W celu określenia błędu konwersji należy przeliczyć liczbę zmiennopozycyjną na postać dziesiętną.

$$m = 1 + 2^{-2} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-11} + 2^{-13} + 2^{-14} + 2^{-15} + 2^{-22} + 2^{-22} =$$

$$= 1.2799999971$$

$$0,01 = (-1)^0 \times 1.2799999971 \times 2^{(120-127)}$$

$$= 0.009999999977648258$$

Liczby zmiennopozycyjne

W standardzie IEEE 754 przewidziano także istnienie zera w dwóch post 0^+ i 0^- . Związane są z sytuacją gdy wynik obliczeń posiada cechę mniejszą od -126. Mówi się wtedy o niedomiarze. (bit znaku odpowiednio 0 lub 1, cecha i mantysa same 0)

Analogiczna sytuacja ma miejsce z nieskończonością w dwóch postaciach $+\infty$ i $-\infty$. Stosuje się je wtedy gdy cecha osiąga wartość większą od 127. (bit znaku odpowiednio 0 lub 1, cecha same 1, mantysa same 0)

Dodatkowo przewidziano także konieczność stosowania symbolu NaN (Not a Number) dla sytuacji gdy działanie jest niewykonalne: $0/0$,

$\infty - \infty$, $x + \text{NaN}$. (bit znaku 0 lub 1, cecha same 1, mantysa dowolna poza samymi 0)

Liczby zmiennopozycyjne

Typ	Bit znaku	Cecha	Mantysa	Wartość
0	0	0000 0000	000 0000 0000 0000 0000 0000	0.0
0 ₋	1	0000 0000	000 0000 0000 0000 0000 0000	-0.0
1	0	0111 1111	000 0000 0000 0000 0000 0000	1.0
-1	1	0111 1111	000 0000 0000 0000 0000 0000	-1.0
Najmniejsza niedomiarowa liczba	*	0000 0000	000 0000 0000 0000 0000 0001	$\pm 2^{-23} \times 2^{-126} = \pm 2^{-149} \approx \pm 1.4 \times 10^{-45}$
Średnia niedomiarowa liczba	*	0000 0000	100 0000 0000 0000 0000 0000	$\pm 2^{-1} \times 2^{-126} = \pm 2^{-127} \approx \pm 5.88 \times 10^{-39}$
Największa niedomiarowa liczba	*	0000 0000	111 1111 1111 1111 1111 1111	$\pm (1 - 2^{-23}) \times 2^{-126} \approx \pm 1.18 \times 10^{-38}$
Najmniejsza liczba znormalizowana	*	0000 0001	000 0000 0000 0000 0000 0000	$\pm 2^{-126} \approx \pm 1.18 \times 10^{-38}$
Największa liczba znormalizowana	*	1111 1110	111 1111 1111 1111 1111 1111	$\pm (2 - 2^{-23}) \times 2^{-127} \approx \pm 3.4 \times 10^{38}$
$+\infty$	0	1111 1111	000 0000 0000 0000 0000 0000	$+\infty$
$-\infty$	1	1111 1111	000 0000 0000 0000 0000 0000	$-\infty$
Wartość nieokreślona	*	1111 1111	Nie same zera	NaN

Liczby zmiennopozycyjne

- Nie każdą liczbę rzeczywistą można przedstawić w postaci maszynowej. W takim wypadku stosując tzw. obcięcie wyznacza się najbliższą jej liczbę maszynową.
- Można zaokrąglić w dół lub w górę.
- Zaokrąglenie w dół powstaje poprzez obcięcie, natomiast zaokrąglenie w górę poprzez zwiększenie ostatniego bitu mantysy.
- Szacując błąd przybliżenia można wybrać dokładniejsze przybliżenie.

Działania na liczbach zmiennopozycyjnych

Działania arytmetyczne (+, -, ×, ÷) wykonywane na maszynach liczących są obarczone błędem wynikającym z dokładności arytmetyki ε , zależną od konstrukcji maszyny liczącej.

Działania wykonane na dwóch liczbach maszynowych obarczone są błędem wynikającym z aproksymacji wyniku obliczeń.

$$fl(x \ominus y) = (x \ominus y)(1 + \delta) \text{ gdzie } |\delta| \leq \varepsilon$$

Działania na liczbach zmiennopozycyjnych

W sytuacji gdy działanie nie dotyczy liczb maszynowych zależność znacznie się komplikuje a błędy aproksymacji implikują.

$$fl(fl(x) \ominus fl(y)) = (x(1+\delta_1) \ominus y(1+\delta_2))(1+\delta_3)$$

gdzie $|\delta_i| \leq \varepsilon$

W współczesnych maszynach liczących działania wykonywane są na specjalnych rejestrach znacznie dłuższych od tych przeznaczonych dla liczb maszynowych

Przykład 4

Wykonać podstawowe działania arytmetyczne na liczbach:

$$x=0.31426 \times 10^3 \quad y=0.92577 \times 10^5$$

$$x+y=0.92891 \text{ 26000} \times 10^5 \quad fl(x+y)=0.92891 \times 10^5$$

$$x-y=-0.92262 \text{ 74000} \times 10^5 \quad fl(x-y)=-0.92263 \times 10^5$$

$$x \times y=0.29093 \text{ 24802} \times 10^8 \quad fl(x \times y)=0.29093 \times 10^8$$

$$x \div y=0.33945 \text{ 79647} \times 10^{-2} \quad fl(x \div y)=0.33946 \times 10^{-2}$$

Przykład 4

Błąd względny aproksymacji obliczeń
wynosi: $\frac{|fl(X) - (X)|}{|(X)|}$

$$\delta(x + y) = \frac{|0.92891 \cdot 10^5 - 0.9289126 \cdot 10^5|}{|0.9289126 \cdot 10^5|} = 2.7989 \cdot 10^{-6}$$

$$\delta(x - y) = 2.81804 \cdot 10^{-6}$$

$$\delta(x \times y) = 8.525 \cdot 10^{-6}$$

$$\delta(x \div y) = 5.99573 \cdot 10^{-6}$$

Błąd względny obliczeń

Błąd względny obliczeń jest równy co najwyżej wielokrotności precyzji obliczeń $n\varepsilon$, gdzie n jest liczbą działań arytmetycznych w wyrażeniu

$$\begin{aligned}
 fl[x \cdot (y + z)] &= [x \cdot fl(y + z)](1 + \delta_1) = & |\delta_1| &\leq \varepsilon \\
 &= [x \cdot (y + z)(1 + \delta_1)](1 + \delta_2) = & |\delta_2| &\leq \varepsilon \\
 &= x \cdot (y + z)(1 + \delta_1 + \delta_2 + \delta_1\delta_2) \approx & |\delta_1\delta_2| &\ll \varepsilon \\
 &\approx x \cdot (y + z)(1 + \delta_1 + \delta_2) = \\
 &= x \cdot (y + z)(1 + \delta_3) = & |\delta_3| &\leq 2\varepsilon
 \end{aligned}$$

Utrata cyfr znaczących

Utrata **cyfr znaczących** (ograniczenie cyfr przechowywanych w mantysie) występuje w przy działaniach algebraicznych.

$$x = 0.3721478693 \quad y = 0.3720230572$$

$$x - y = 0.0001248121$$

$$fl(x) = 0.37215 \quad fl(y) = 0.37202$$

$$fl(x) - fl(y) = 0.00013 = 0.13000 \times 10^{-3}$$

$$\left| \frac{(x - y) - (fl(x) - fl(y))}{(x - y)} \right| = \left| \frac{0.0001248121 - 0.00013}{0.00013} \right| \approx 0.04$$

Odejmowanie liczb bliskich

Programując zadania obliczeniowe należy unikać sytuacji w których pojawi się ryzyko odejmowania liczb bliskich.

$$y = \sqrt{x^2 + 1} - 1$$

$$y = \left(\sqrt{x^2 + 1} - 1 \right) \frac{(\sqrt{x^2 + 1} + 1)}{(\sqrt{x^2 + 1} + 1)} = \frac{x^2}{(\sqrt{x^2 + 1} + 1)}$$

W pierwszym równaniu dla małych **x** zachodzi odejmowanie liczb bliskich i ryzyku utraty cyfr znaczących w wyniku.

Twierdzenie

Jeżeli liczby maszynowe są takie, że

$$x > y > 0 \quad i \quad 2^{-q} \leq 1 - \frac{x}{y} \leq 2^{-p}$$

gdzie **p** i **q** są liczbami całkowitymi, to liczba bitów znaczących straconych przy odejmowaniu **x – y** jest równa co najmniej **p** i co najwyżej **q**.

Pojawia się więc w wyniku co najmniej **p** i co najwyżej **q** obcych zer, co oznacza utratę odpowiednio **p** lub **q** bitów znaczących.

Arytmetyka przedziałowa

- Rozwiązaniem problemu utraty cyfr znaczących wydaje się stosowanie arytmetyki przedziałowej.
- Zamiast na liczbach operuje się w niej na przedziałach zawierających dokładne wartości.
- W przypadku idealnym operuje się na bardzo małych przedziałach.
- Ze względu na duże koszty działania oraz problemem rozszerzania się tworzonych przedziałów, metodę tą stosuje się tylko w sytuacjach gdy dokładne informacje o wynikach mają szczególne znaczenie.

Stabilność algorytmów

Algorytm uznawany jest za stabilny wtedy gdy rozwiązanie zadania jakie generuje obarczone jest błędem tego samego rzędu co optymalny poziom błędu rozwiązania danego zadania.

Oznacza to że otrzymywany z algorytmu błąd jest na poziomie błędu wynikającego z przybliżonej reprezentacji danych i wyniku.

Stabilność algorytmów

Algorytmem niestabilnym określić można algorytm numeryczny w którym niewielkie błędy popełniane na jakimś etapie przenoszą się i rosną na kolejnych etapach obliczeń, powodując znaczące zniekształcenie ostatecznego wyniku obliczeń.

Przykład 5

Sprawdzić stabilność algorytmu rekurencyjnego obliczanego w arytmetyce zmiennopozycyjnej o mantysie 24 cyfrowej.

$$x_0 = 1, \quad x_1 = \frac{1}{3}, \quad x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1}$$

Postać ogólna równania ma postać:

$$x_0 = a_0, \quad x_1 = b_1$$

$$x_{n+1} = \alpha x_n + \beta x_{n-1}, \quad \text{dla } n \geq 1$$

Równanie charakterystyczne:

$$x^2 - \alpha x - \beta = 0$$

Przykład 5

Wyznaczenie wzoru ogólnego ciągu realizujemy korzystając z równania charakterystycznego według wzoru:

$$\Delta > 0, x_n = Ar_1^n + Br_2^n$$

$$x_0 = A + B$$

$$x_1 = Ar_1 + Br_2$$

$$\Delta = 0, x_n = Ar^n + Bnr^n$$

$$x_0 = A$$

$$x_1 = Ar + Br$$

Przykład 5

$$x^2 - \frac{13}{3}x + \frac{4}{3} = 0$$

$$\Delta = b^2 - 4ac = \left(-\frac{13}{3}\right)^2 - 4 \cdot 1 \cdot \frac{4}{3} = \frac{169}{9} - \frac{16}{3} =$$

$$= \frac{169 - 16 \cdot 3}{9} = \frac{121}{9}, \quad \sqrt{\Delta} = \sqrt{\frac{121}{9}} = \frac{11}{3}$$

Przykład 5

$$r_1 = \frac{-b - \sqrt{\Delta}}{2a} = \frac{\frac{13}{3} - \frac{11}{3}}{2} = \frac{2}{3} \cdot \frac{1}{2} = \frac{1}{3}$$

$$r_2 = \frac{-b + \sqrt{\Delta}}{2a} = \frac{\frac{13}{3} + \frac{11}{3}}{2} = \frac{24}{3} \cdot \frac{1}{2} = \frac{12}{3} = 4$$

$$1 = A + B, \quad \frac{1}{3} = A \frac{1}{3} + 4B$$

$$A = 1, \quad B = 0 \rightarrow x_n = Ar_1^n + Br_2^n = \left(\frac{1}{3}\right)^n$$

Przykład 5

$$\begin{aligned}
 x_{m+1} &= \frac{13}{3}x_m - \frac{4}{3}x_{m-1} = \frac{13}{3}\left(\frac{1}{3}\right)^m - \frac{4}{3}\left(\frac{1}{3}\right)^{m-1} = \\
 &= \left(\frac{1}{3}\right)^{m-1} \left(\frac{13}{3} \frac{1}{3} - \frac{4}{3}\right) = \left(\frac{1}{3}\right)^{m-1} \left(\frac{13}{9} - \frac{4 \cdot 3}{9}\right) = \\
 &= \left(\frac{1}{3}\right)^{m-1} \left(\frac{13}{9} - \frac{12}{9}\right) = \left(\frac{1}{3}\right)^{m-1} \left(\frac{1}{9}\right) = \left(\frac{1}{3}\right)^{m-1} \left(\frac{1}{3}\right)^2 = \\
 &= \left(\frac{1}{3}\right)^{m-1+2} = \left(\frac{1}{3}\right)^{m+1}
 \end{aligned}$$

**Sprawdzenie zgodność
wzory na n -ty element ciągu z
warunkami początkowymi**

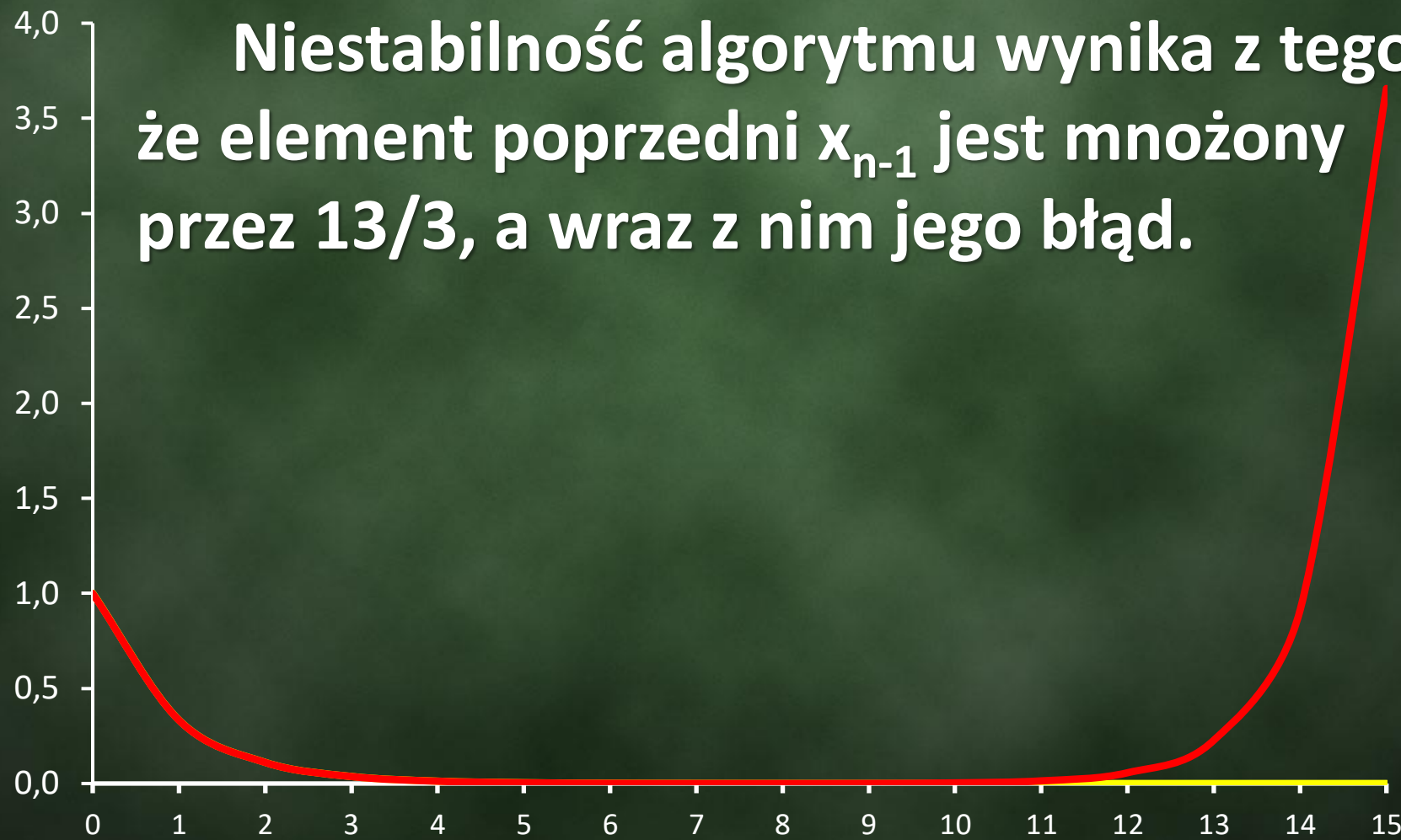
Przykład 5

Wyznaczając kolejne elementy ciągu x_n w arytmetyce zmiennopozycyjnej z 24 cyfrowej mantysie:

x_0	1.00000 00	x_8	0,00037 57 (0 cyfr znaczących)
x_1	0.33333 33 (7 cyfr znaczących)	x_9	0,00094 57
x_2	0,11111 12 (6 cyfr znaczących)	x_{10}	0,00358 87
x_3	0,03703 73 (5 cyfr znaczących)	x_{11}	0,01429 27
x_4	0,01234 66 (4 cyfr znaczących)	x_{12}	0,05715 02
x_5	0,00411 87 (3 cyfr znaczących)	x_{13}	0,22859 39
x_6	0,00138 57 (2 cyfr znaczących)	x_{14}	0,91437 35
x_7	0,00051 31 (1 cyfr znaczących)	x_{15}	3,65749 3 (błąd względny 10^8)

Przykład 5

Niestabilność algorytmu wynika z tego że element poprzedni x_{n-1} jest mnożony przez $13/3$, a wraz z nim jego błąd.



Stabilność algorytmu

W niektórych sytuacjach pomimo, że błędy bezwzględne są duże, ale z powodu osiąganych przez algorytm dużych wartości są przez nie „zagłuszone”, traktuje się je jako pomijalnie względne.

$$x_0 = 1, x_1 = 4, x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1}$$

$$x_n = 4^n$$

$$x_1 = 4.00000\ 60, x_{10} = 1.04857\ 6 \cdot 10^6, x_{20} = 1.09951\ 2 \cdot 10^{12}$$

Uwarunkowanie zadania

- Pod pojęciem uwarunkowania zadania numerycznego rozumie się wrażliwość rozwiązania na dane wejściowe.
- Zadanie źle uwarunkowane cechuje się dużymi względnymi zmianami wyników na małe względne zmiany danych wejściowych.
- Niezależnie od obranej metody obliczeniowej zadania źle uwarunkowanego wyniki uzyskane w rozwiązaniu obarczone są dużymi błędami.

Wskaźnik uwarunkowania

- Wskaźnik uwarunkowania **K** określa w jakim stopniu błąd danych wejściowych wpływa na błąd wyniku.
- Wskaźnik uwarunkowania definiuje się jako maksymalny stosunek błędu względnego rozwiązania do błędu względnego danych.
- Algorytm o niskim wskaźniku uwarunkowania nazywamy dobrze uwarunkowanym
- Algorytm o wysokim wskaźniku uwarunkowania – źle uwarunkowanymi.

Wskaźnik uwarunkowania

- Zagadnienia o zbyt dużym wskaźniku uwarunkowania nie nadają się do numerycznego rozwiązywania ponieważ, już sam błąd wynikający z numerycznej reprezentacji liczb wprowadza nieproporcjonalnie duży błąd.

$$\delta_f = \frac{f(x+h) - f(x)}{f(x)} \approx \frac{hf'(x)}{f(x)} = \frac{hf'(x)}{f(x)} \cdot \frac{x}{x} = \frac{xf'(x)}{f(x)} \cdot \frac{h}{x}$$

$$\delta_f = \frac{xf'(x)}{f(x)} \cdot \delta_x = \kappa \cdot \delta_x$$

Przykład 6

Określić wskaźnik uwarunkowania funkcji **arcsin(x)**. $f(x) = \arcsin(x), f'(x) = \frac{1}{\sqrt{1-x^2}}$

$$\kappa = \frac{x \cdot f'(x)}{f(x)} = \frac{x}{\sqrt{1-x^2} \cdot \arcsin(x)}$$

Dla x bliskich 1 $\arcsin(x) \approx \pi/2$ powoduje to że wskaźnik uwarunkowania jest bardzo duży.

W efekcie bardzo małe względne błędy wartości wejściowej $x \approx 1$ powodują duże błędy względne $\arcsin(x)$.