

## I. Cel ćwiczenia

Celem ćwiczenia laboratoryjnego jest zapoznanie się z oprogramowaniem narzędziowym, dedykowanym do wykonywania zaawansowanych obliczeń numerycznych. Część pierwsza dotyczy zagadnienia podstaw posługiwania się przedmiotowym oprogramowaniem (uzyskanie podstawowych informacji o programie, jego instalacji, sposobach wydawania poleceń oraz uruchamiania skryptów itp.).

## II. Dostępne narzędzia programowe

W laboratorium wykorzystywany jest darmowy program **Scilab** (<http://www.scilab.org/>), należący do grupy oprogramowania *open source*, który został opracowany w 1990 roku przez francuskich uczonych z *Institut National de Recherche en Informatique et en Automatique*. Od 1994 roku program jest dystrybuowany przez Internet, natomiast od maja 2003 roku aplikacja jest rozwijana przez „*Scilab Consortium*”, składające się z 18 instytucji naukowych i przemysłowych. Scilab jest programem naukowym, dedykowanym do wykonywania obliczeń numerycznych. Program stanowi ograniczoną funkcjonalnie, ale całkowicie bezpłatną alternatywę dla popularnego Matlab’a. Scilab posiada kilkaset funkcji matematycznych i pozwala na operowanie nie tylko na liczbach, ale także na zaawansowanych strukturach, przykładowo takich jak macierze lub wektory. Aplikacja umożliwia prezentowanie wyników obliczeń na zaawansowanych grafach oraz dwu i trójwymiarowych wykresach. W programie wbudowano interpreter oraz język programowania wysokiego poziomu. Opracowywane przez użytkownika programy (nazywane skryptami) nie są kompilowane przed ich wykonaniem. Wprowadzane z klawiatury, lub odczytywane z pliku polecenia, są na bieżąco interpretowane w trakcie realizacji programu. Scilab może być wykorzystywany zarówno do podstawowych, jak i zaawansowanych obliczeń numerycznych, przykładowo obejmujących obszar: algebry liniowej, analizy sygnałów fraktalnych, grafów i sieci, macierzy, modeli Markowa, modelowania i symulacji układów dynamicznych, przetwarzania obrazów i sygnałów, sieci neuronowych, statystyki, strategii ewolucyjnej, szeregów czasowych i wielu innych aplikacji, które znajdują zastosowanie w różnych obszarach nauki i techniki.

## III. Literatura

- *Scilab manual*, <http://www.scilab.org/>.
- A. Brozi, *Scilab w przykładach*, NAKOM, Poznań 2007.
- C. T. Lachowicz, *Matlab, Scilab, Maxima. Opis i przykłady zastosowań*, Wydawnictwo Politechniki Opolskiej, Opole 2005.
- S. L. Campbell, J.-P. Chancelier, R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos*, Springer, 2005.
- B. Pinçon, *Wprowadzenie do Scilaba*, Université Henri Poincaré, przekład z języka francuskiego.

## IV. Realizacja ćwiczenia laboratoryjnego

**Uwaga:** w czasie realizacji poszczególnych części ćwiczenia laboratoryjnego należy wspomagać się informacjami, które zawarto w podręcznej pomocy programu Scilab, uruchamianej klawiszem *F1*, niezależnie dla każdego aktywnego okna programu, lub rozkazem **help nazwa\_instrukcji**, wprowadzaną (po znaku zachęty **-->**) w oknie głównym konsoli.

1. Należy uruchomić program Scilab i zapoznać się z przeznaczeniem podstawowych części aplikacji:
  - okno główne konsoli (*Konsola Scilab*);
  - okno edytora skryptów *SciNotes* – menu */Applications/SciNotes*;
  - okno narzędzia do modelowania i symulacji układów dynamicznych – menu */Applications/Xcos* (szczegółowe informacje na temat narzędzia znajdują się na stronie <http://www.scicos.org>);
  - okno narzędzia do konwersji skryptów Matlab'a na Scilab'a – menu */Applications/Translator Matlab do Scilab*;
  - okno przewodnika umożliwiającego zapoznanie się z możliwościami oprogramowania – menu */Przykłady* (po wybraniu interesującej sekcji przewodnika, dla niektórych sekcji można przełączyć się na okno konsoli i - np. za pomocą klawisza *Enter* – uruchamiać kolejne elementy wybranego skryptu demonstracyjnego, jeżeli są one dostępne).
2. Wykonując podstawowe operacje matematyczne na liczbach (np. dodawanie **+**, odejmowanie **-**, mnożenie **\***, dzielenie **/**, **\** - realizowane podobnie jak na kalkulatorze), przeprowadzić testy pracy programu w **trybie poleceń**. Rozkazy należy wpisywać w oknie konsoli, po znaku zachęty **-->**. Polecenia są wykonywane natychmiast po naciśnięciu klawisza *Enter*. Przetestować sposoby komunikacji z programem, które zostały zawarte w poniższych uwagach.

### Uwagi praktyczne dotyczące komunikacji z programem

- 2.1. Operatory arytmetyczne nie muszą być oddzielane od zmiennych spacjami, chociaż – w szczególnych przypadkach – spacje pozwalają uniknąć niejednoznaczności podczas definiowania zmiennych.
- 2.2. Program zapamiętuje pewną liczbę wcześniej wydanych rozkazów, których wybieranie można realizować za pomocą klawiszy strzałek **↑** **↓**. Naciśnięcie klawisza **↑** powoduje wyświetlenie ostatnio wydanego rozkazu, kolejne jego naciśnięcia przywołują wyświetlanie wcześniejszych rozkazów. Historię można wykasować za pomocą opcji menu */Preferences/Wyczyść historię*. Opcja menu */Preferences/Wyczyść konsolę* czyści okno konsoli.
- 2.3. W sytuacji konieczności zapisu polecenia w kilku wierszach, np. dla zwiększenia przejrzystości zapisu matematycznego, należy wykorzystać znak dwóch kropek **..** (po zastosowaniu tego znaku, naciśnięcie klawisza *Enter* spowoduje przejście do nowej linii, a program będzie czekał na dokończenie polecenia w kolejnej linii lub liniach, jeżeli one także będą zakończone tym symbolem). Metody tej nie należy stosować do przenoszenia do następnego wiersza części wprowadzanego łańcucha.
- 2.4. W celu uniknięcia wyświetlania wyniku na ekranie (np. dla definiowanych, obszernych macierzy), wprowadzane polecenie należy zakończyć symbolem średnika **;**. W trakcie wpisywania kilku poleceń, można je zapisać w jednej linii, oddzielając kolejne elementy średnikami (wyniki nie będą wyświetlane na ekranie) lub przecinkami (wyniki będą wyświetlane na ekranie).
- 2.5. Wprowadzanie wielu poleceń w jednej linii może być kłopotliwe w razie napotkania błędu. W przypadku zaistnienia takiej sytuacji Scilab wyświetli numer wiersza, w którym wystąpił błąd, nie będzie jednak wiadomo, w którym konkretnie poleceniu on wystąpił. Niewygodne może być także korzystanie z możliwości wprowadzenia jednego polecenia w wielu liniach, separowanych za pomocą symbolu **..**. Takie wyrażenia traktowane są jako pojedyncza linia i odpowiednio do tego zostanie podany numer ewentualnego wiersza, w którym wystąpił błąd.
- 2.6. Sposób zapisu liczb (dokładność): Do zmiany formatu wyświetlania (i drukowania) liczb służy funkcja **format ([typ],[długość])**, gdzie **typ**'em może być symbol **'e'** (exponential - postać wykładnicza) lub **'v'** (variable - format zmienny, ustawiony domyślnie); domyślna **długość** wynosi 10.

- 2.7. Podczas wykonywania skryptu, naciśnięcie kombinacji *Ctrl-C* (znak zachęty zmieni się z `-->` na `-n->`, gdzie **n** oznacza krotność użycia kombinacji klawiszy *Ctrl-C*) powoduje przerwanie aktualnie wykonywanych operacji (odczytanych ze skryptu lub poleceń wprowadzonych z klawiatury) i przekazanie kontroli użytkownikowi. Ten sam efekt uzyskiwany jest za pomocą opcji menu */Sterowanie/Przerwij*. W tym stanie dostępne są wszystkie aktualnie znajdujące się w pamięci zmienne, które można oglądać i/lub zmieniać. Po wydaniu polecenia **resume** Scilab powraca do wykonywania programu, z uwzględnieniem nowo wprowadzonych zmian.
3. Zapoznać się ze sposobem definiowania i korzystania ze zmiennych (także systemowych – patrz uwagi poniżej). Napisać skrypt, w którym dla czterech zdefiniowanych zmiennych  $x_1=0^\circ$ ,  $x_2=45^\circ$ ,  $x_3=90^\circ$  i  $x_4=180^\circ$ , zostaną wyliczone funkcje **sin(x)** i **cos(x)**. W skrypcie należy wykorzystać zmienną systemową **%pi**. Przetestować działanie funkcji **clear**.

### Uwagi praktyczne dotyczące definiowania zmiennych

- 3.1. Operator przypisania `=` oznacza, że wartość wyrażenia po jego prawej stronie zostaje przypisana zmiennej znajdującej się z lewej strony. Długość nazw zmiennych jest nieograniczona, lecz w Scilab'ie rozpoznawane są tylko 24 pierwsze znaki. Nazwy muszą zaczynać się od litery lub znaku `%`, po których następują dowolne znaki - litery, cyfry lub inne elementy (poza spacją i symbolami operatorów).
- 3.2. Przy pracy z programem użyteczna jest rozkaz **clear**, która powoduje usuwanie z pamięci zmiennych, wskazywanych jako jej parametry. Parametry tego rozkazu podaje się po spacji, a jeśli jest ich więcej - oddziela się je od siebie spacjami. Rozkaz ten usuwa wszystkie zmienne z pamięci, jeżeli wywołano ją bez parametrów.
- 3.3. Zmienne o nazwach zaczynających się od znaku `%`, są **zmiennymi systemowymi** (predefiniowanymi), np.: **%eps**, **%t** lub **%pi**. Rozkaz **clear** nie kasuje tych zmiennych. Użytkownik może zdefiniować własne zmienne o nazwach zaczynających się od znaku `%`, jednak taka nazwa nie czyni ich odpornymi na działanie polecenia **clear**.

Użyteczne zmienne systemowe:

- wartość  $\pi$  – symbol **%pi**;
- podstawa logarytmów naturalnych  $e$  - symbol **%e**;
- jednostka urojona liczby zespolonej - symbol **%i** (na ekranie pojawia się tylko **i**);
- nieskończoność - symbol **%inf**;
- prawda i fałsz logiczny - symbol **%t** lub **%T**, oraz: **%f** lub **%F**;
- minimalna możliwa do zapisania w danym systemie operacyjnym liczba zmiennoprzecinkowa - symbol **%eps** (można ją wykorzystać np. do zatrzymania obliczeń gdy krok zmaleje do praktycznie „niemierzalnej” wartości danej zmiennej);
- „nie liczba” - symbol **%nan**;
- zmienna specjalna – symbol **ans** (do tej zmiennej przypisywany jest wynik ostatniego polecenia, jeśli polecenie to samo tego nie określało; przykład: po wprowadzeniu polecenia **3\*2** zmienna **ans=6**; zmienna zachowuje obliczoną wartość dopóki nie zostanie wydane kolejne tego rodzaju polecenie).

### 3.4. Wybrane, użyteczne funkcje programu:

abs	wartość bezwzględna, moduł
exp	eksponent
log	logarytm naturalny
log10	logarytm o podstawie 10
cos	cosinus (argument w radianach)
sin	sinus (argument w radianach)
sinc	$\frac{\sin(x)}{x}$
tan	tangente (argument w radianach)
cotg	cotangente (argument w radianach)
acos	arccos
asin	arcsin
atan	arctg
cosh	cosinus hiperboliczny
sinh	sinus hiperboliczny
tanh	tangens hiperboliczny
acosh	argch
asinh	argsh
atanh	argth
sqrt	pierwiastek kwadratowy
floor	$E(x) = (\lfloor x \rfloor) = n \Leftrightarrow n \leq x < n + 1, \quad x \in N$
ceil	$\lceil x \rceil = n \Leftrightarrow n - 1 < x \leq n, \quad x \in N$
int	$\text{int}(x) = \lfloor x \rfloor$ jeśli $x > 0$ oraz $\lceil x \rceil$ dla $x \leq 0$
erf	funkcja błędu $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$
erfc	dopełnienie funkcji błędu określone przez $\text{erfc}(x) = 1 - \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt$
gamma	$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$
lngamma	$\ln(\Gamma(x))$
dlgamma	$\frac{d}{dx} \ln(\Gamma(x))$

4. Przeprowadzić operacje na macierzach korzystając **ze wszystkich** poniższych uwag.

#### Uwagi praktyczne dotyczące operacji na macierzach

4.1. Podstawowym typem danych w Scilab'ie jest **macierz** liczb rzeczywistych lub zespolonych. Wektor lub skalar są w istocie szczególnymi przypadkami macierzy. Najprostszym sposobem definiowania macierzy w programie jest wprowadzenie z klawiatury listy jej elementów według poniższej zasady:

- elementy tego samego wiersza oddzielone są spacją lub przecinkiem;
- lista elementów musi być ujęta w nawias kwadratowy;
- każdy wiersz macierzy (z wyjątkiem ostatniego) musi być zakończony średnikiem.

### Przykłady definicji macierzy (wektorów):

- wektor wierszowy:  $\mathbf{A} = [1, 2, 3]$  lub  $\mathbf{A} = [1 \ 2 \ 3]$ ;
- wektor kolumnowy:  $\mathbf{A} = [1; 2; 3]$ ;
- macierz o rozmiarze 2 wierszy i 3 kolumn (lub inaczej 2x3):  $\mathbf{A} = [1, 2, 3; 4, 5, 6]$  lub inaczej, z użyciem symbolu  $\dots$ :  
$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$
;
- macierz transponowana:  $\mathbf{A}'$ ;
- macierz o elementach losowych z wykorzystaniem funkcji **rand**, która pozwala zdefiniować grupę elementów pseudolosowych, pochodzących z przedziału [0,1); dla funkcji **rand** możliwe jest także użycie rozkładu normalnego dla losowanych liczb, a także podanie zarodka dla generatora liczb pseudolosowych – szczegółowe informacje zawarto w pomocy; przykład definicji:  $\mathbf{A} = \text{rand}(3, 4)$ ;

### 4.2. Przykłady wyboru elementu/ów macierzy:

- wybór elementu z pierwszej wiersza drugiej kolumny:  $\mathbf{A}(1, 2)$ ;
- równoważny wybór całego drugiego wiersza macierzy o rozmiarze 2x3:  $\mathbf{A}(2, 1:3)$ ,  $\mathbf{A}(2, 1:\$)$ ,  $\mathbf{A}(2, :)$ ;
- wybór fragmentu macierzy od punktu o współrzędnych 1x2 do punktu 2x3:  $\mathbf{A}(1:2, 2:3)$ .

### 4.3. Użyteczne funkcje dotyczące macierzy (wektorów):

- macierz wypełniona zerami: **zeros(4, 3)**;
- macierz wypełniona jedynkami: **ones(4, 3)**;
- macierz jednostkowa: **eye(4, 4)**;
- funkcja zwraca rozmiar macierzy: **size(A)**;
- funkcja zwraca liczbę elementów macierzy: **length(A)**;
- funkcja zwraca ślad macierzy (sumę elementów głównej przekątnej macierzy kwadratowej): **trace(A)**;
- funkcja zwraca wyznacznik macierzy: **det(A)**;
- funkcja zwraca sumę elementów macierzy: **sum(A)**;
- funkcja zwraca iloczyn elementów macierzy: **prod(A)**.

### 4.4. Podstawowe działania na macierzach:

- suma macierzy wg definicji  $\mathbf{A} + \mathbf{B} = [a_{i,j} + b_{i,j}]$ :  $\mathbf{A} + \mathbf{B}$ ;
- mnożenie macierzy przez liczbę wg definicji  $c * \mathbf{B} = [c \cdot b_{i,j}]$ :  $c * \mathbf{B}$ ;
- iloczyn Cauchy'ego  $\mathbf{A} * \mathbf{B}$  wg definicji 
$$c_{i,j} = \sum_{r=1}^m a_{i,r} \cdot b_{r,j} = a_{i,1} \cdot b_{1,j} + a_{i,2} \cdot b_{2,j} + \dots + a_{i,m} \cdot b_{m,j}$$
 dla  $i=1..n$  oraz  $j=1..p$ .

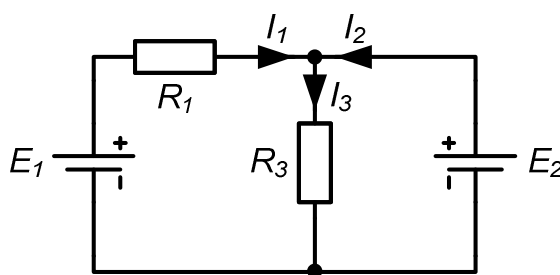
UWAGA: działanie to zdefiniowane jest wyłącznie dla macierzy, z których pierwsza ma tyle kolumn, co druga wierszy. Powyższa definicja jest słuszna dla macierzy  $\mathbf{A}$  o rozmiarze  $n \times m$ , i macierzy  $\mathbf{B}$  o rozmiarze  $m \times p$ , dla których iloczyn  $\mathbf{A} * \mathbf{B}$  jest macierzą  $\mathbf{C}$  o rozmiarze  $n \times p$ .

- iloczyn skalarny wg definicji  $\mathbf{A} \cdot \mathbf{B} = [a_{i,j} \cdot b_{i,j}]$ :  $\mathbf{A} \cdot \mathbf{B}$ ;
- potęgowanie macierzy kwadratowych: np.  $\mathbf{A}^2$ ;  $\mathbf{A}^3$  itd.;
- potęgowanie elementów macierzy:  $\mathbf{A} \cdot ^2$ ;  $\mathbf{A} \cdot ^3$  itd.;
- odwracanie macierzy: **inv(A)** to samo działanie zapisane inaczej:  $\mathbf{A}^{-1}$ ;
- iloraz lewostronny macierzy:  $\mathbf{C} = \mathbf{A} / \mathbf{B}$  co oznacza to samo dla zapisu  $\mathbf{A} = \mathbf{C} * \mathbf{B}$  (uwaga – patrz iloczyn Cauchy'ego);

- iloraz prawostronny macierzy:  $C=A \setminus B$ , co oznacza to samo dla zapisu:  $B=A * C$  (uwaga j.w.);
- skalarny iloraz lewo- i prawostronny:  $C=A ./ B$  oraz  $C=A ./ B$  (służący do dzielenia przez siebie odpowiednich elementów macierzy).

5. Korzystając z okna edytora skryptów (menu konsoli: */Applications/SciNotes*) napisać program, za pomocą którego zostaną wyznaczone prądy  $I_1$ ,  $I_2$  oraz  $I_3$  dla układu z poniższego schematu. Przygotowany skrypt należy zapisać w katalogu *D:\LABORATORIUM\nazwa\_przedmiotu\nazwa\_roku\nr\_gr\_lab\*, w postaci pliku z rozszerzeniem *\*.sce* (domyślne rozszerzenie pliku skryptu w programie Scilab). Wykonanie skryptu można zrealizować opcją menu: */Wykonaj/...plik*, z *echem Ctrl+L*. Każdy skrypt warto rozpocząć od następujących rozkazów:

- **clear** – czyszczenie pamięci;
- **clc** – czyszczenie konsoli;
- **clf** – czyszczenie okna graficznego (jeżeli korzysta się z kreślenia wykresów).



Dane dla poszczególnych zespołów laboratoryjnych (każdy zespół rozwiązuje zadanie dla jednego zestawu danych):

- zespół nr 1:  $E_1=5$  V,  $E_2=1$  V,  $R_1=3$   $\Omega$ ,  $R_3=4$   $\Omega$ ;
- zespół nr 2:  $E_1=10$  V,  $E_2=1$  V,  $R_1=6$   $\Omega$ ,  $R_3=5$   $\Omega$ ;
- zespół nr 3:  $E_1=3$  V,  $E_2=8$  V,  $R_1=2$   $\Omega$ ,  $R_3=3$   $\Omega$ ;
- zespół nr 4:  $E_1=8$  V,  $E_2=3$  V,  $R_1=4$   $\Omega$ ,  $R_3=4$   $\Omega$ .

### 5.1. Sformułowanie problemu

W pierwszej kolejności, układ elektryczny z powyższego schematu należy opisać układem liniowych równań postaci:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \end{cases}$$

W zapisie macierzowym układ ten przyjmuje postać  $A \cdot x = b$ :

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Rozwiązaniem układu równań postaci  $A \cdot x = b$  jest rozkaz wykorzystujący:

- macierz odwrotną, np:  $x1 = \text{inv}(A) * b$ ;
- funkcję **linsolve**, która rozwiązuje równania macierzowe postaci  $A \cdot x + b = 0$ ; rozwiązaniem układu równań jest przykładowy rozkaz:  $x2 = \text{linsolve}(A, -b)$ .

## 5.2. Analiza numerycznego rozwiązania przybliżonego

Uzyskane rozwiązanie numeryczne - pomimo dość dokładnego wyniku - jest zawsze obarczone pewnymi błędami zaokrąglenia, spowodowanymi arytmetyką liczb zmiennoprzecinkowych. W momencie, gdy rozwiązanie układu liniowego jest wątpliwe, Scilab wyświetla stosowne ostrzeżenia (*Warnings*), pozwalające podjąć użytkownikowi odpowiednie działania.

Wykonując rozkazy: `r1=A*x1-b` oraz `r2=A*x2-b` sprawdzić, czy obydwa uzyskane rozwiązania (wykorzystujące macierz odwrotną i funkcję `linsolve`) są dokładne. Rozwiązanie dokładne oznacza spełnienie macierzowego równania postaci:  $A \cdot x - b = 0$ . Na podstawie uzyskanych wyników należy ocenić, które rozwiązanie jest bardziej dokładne.

Przy ocenie dokładności obliczeń dodatkowo należy posłużyć się normą macierzową, która określa największą wartość bezwzględną elementu macierzy. W tym celu należy wykonać następujące rozkazy: `n1=norm(A*x1-b, 'inf')`, `n2=norm(A*x2-b, 'inf')`, a następnie porównać i skomentować uzyskane wyniki.

## IV. Sprawozdanie z ćwiczenia laboratoryjnego

**Uwaga:** sprawozdanie – przygotowane w pliku \*.docx (\*.doc) na podstawie wszystkich wytycznych formatki *mn\_formatka.doc* – należy przesłać na adres [pjanko@prz.edu.pl](mailto:pjanko@prz.edu.pl), **najpóźniej w terminie 1 tygodnia od dnia zakończenia ćwiczenia laboratoryjnego.** Odbiór każdego sprawozdania zostanie potwierdzony wiadomością zwrotną przez prowadzącego zajęcia. Sprawozdania powinny być opisane: nazwa roku, datą wykonania ćwiczenia, numerem grupy laboratoryjnej, numerem zespołu ćwiczącego (1-4) i jego składem osobowym. W przypadku odrabiania ćwiczenia, w/w informacje powinny zostać zawarte przy nazwisku osoby odrabiającej laboratorium. Odpowiedzialność za sprawozdanie jest zbiorowa, co oznacza, że wszyscy członkowie danego zespołu ćwiczącego (1-4) otrzymują tą samą ocenę za złożone sprawozdanie.

Sprawozdanie z pierwszego ćwiczenia laboratoryjnego powinno zawierać kompleksowe rozwiązanie zadania z punktu 5. W sprawozdaniu należy dokonać analitycznego i numerycznego rozwiązania zadanego problemu, stosując obszerny komentarz do każdego etapu postępowania. W dokumencie należy zawrzeć skrypt z komentarzem do każdej linii programu (komentarz w skrypcie wykonuje się po znaku `//`), a także opis przeprowadzonej analizy przybliżonego rozwiązania numerycznego. Sprawozdanie należy podsumować wnioskami, wyciągniętymi z realizacji procesu obliczeń numerycznych w programie Scilab.

## V. Przygotowanie do następnych zajęć

1. Wiedza teoretyczna z zakresu rachunku macierzowego.
2. Umiejętność posługiwania się oprogramowaniem narzędziowym w zakresie zrealizowanym podczas pierwszego ćwiczenia laboratoryjnego.