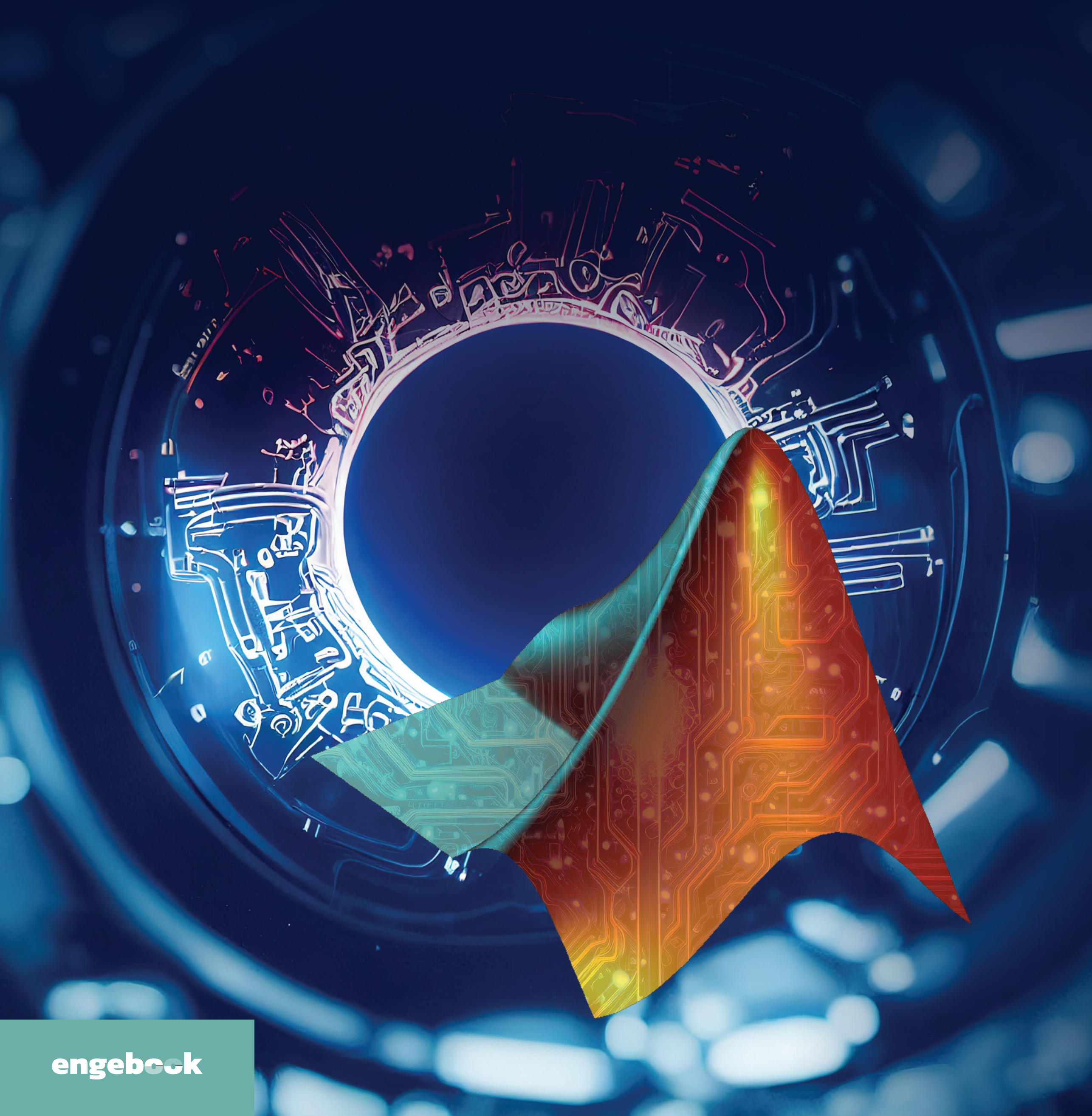
MATLAB: FROM FUNDAMENTALS TO ARTIFICIAL INTELLIGENCE

FILIPE AZEVEDO



engebook

AUTHOR
Filipe Azevedo

TITLE
MATLAB: FROM FUNDAMENTALS TO ARTIFICIAL INTELLIGENCE

PUBLISHER
Quântica Editora – Conteúdos Especializados, Lda.
Tel. +351 220 939 053 · Email: geral@quanticaeditora.pt · www.quanticaeditora.pt
Praça da Corujeira no. 38 · 4300-144 PORTO · Portugal

IMPRINT
Engebook – Conteúdos de Engenharia

DISTRIBUTION
Booki – Conteúdos Especializados
Tel. +351 220 104 872 · Email: info@booki.pt · www.booki.pt

COVER DESIGN
Luciano Carvalho
Delineatura, Design de Comunicação · www.delineatura.pt

PRINTING
July 2025

This book complies with the new Orthographic Agreement of 1990, following its general guidelines and assuming some specific options.

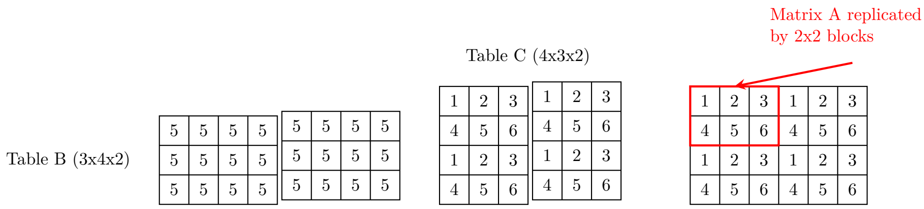
UDC (Universal Decimal Classification)

004.4	Software. Equipment, programs
004.8	Artificial intelligence
519.6	Computational mathematics. Numerical analysis and mathematical programming. Mathematics for engineering
004.45	System software
62	Engineering. Technology in general

ISBN
E-book: 9789899177871

Publication cataloging
Family: Engineering
Subfamily: Programming Languages

Contents



(a) Matrix of 2 pages each with dimension 3x4

(b) Two pages of replicas of A

(c) Matrix A replicated by blocks

Figure 3.8: Pre-allocation of matrices.

3.6 Concatenation of matrices

a) Using square brackets

Ex:
```
A = [1 2; 3 4];
B = [5 6; 7 8];
C = [A, B];         %Horizontal concatenation
D = [A; B];         %Vertical concatenation
```

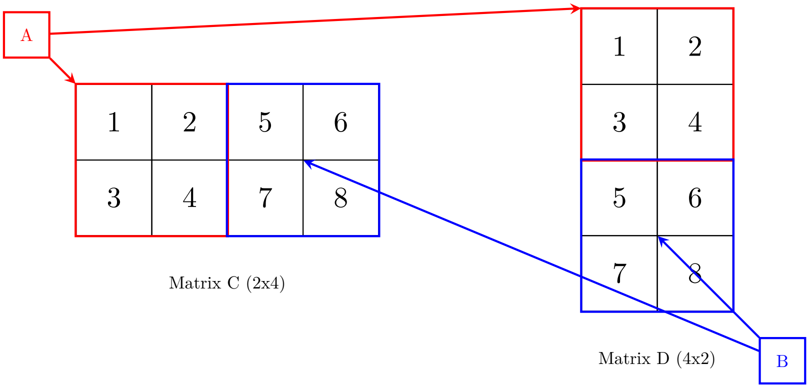


Figure 3.9: Concatenation of matrices.

b) Functions `horzcat` and `vertcat`

```
horzcat(A,B,C, ...)
```

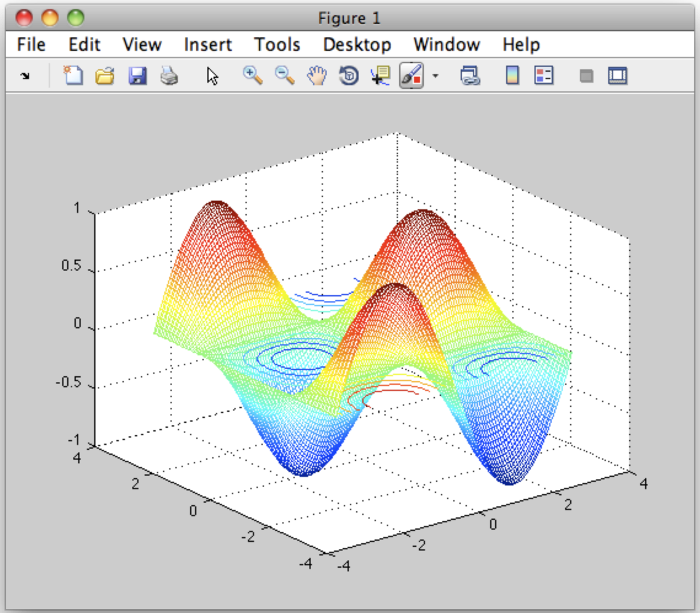


Figure 4.13: Functions `mesh` and `contour` in the same figure

4.6 Function `subplot`

The `subplot` function allows you to construct multiple graphs in the same window by dividing it into rows and columns. Consider the following example:

```
>>figure(1) % Creates the figure window
>>subplot(2,1,1) % Divides the figure window into 2 rows and 1 column
                 % and constructs the graph in position (1,1)
>>contour(X,Y,Z);
>>subplot(2,1,2) % Constructs the graph in position (2,1) of the
    figure
                 % window
>>mesh(X,Y,Z)
```

The output is displayed in Figure 4.14.

11.9 Advanced Applications and Real-Time Processing

11.9.1 Real-Time Processing

MATLAB allows simulating real-time processing, useful for prototyping acquisition and control systems.

```
while true
    audio = getaudiodata(recObj);
    plot(audio); drawnow;
    pause(0.1);
end
```

11.9.2 Development of Graphical Interfaces for Signals

```
function signalViewer
    f = figure('Name','Signal Viewer');
    ax1 = subplot(2,1,1); ax2 = subplot(2,1,2);
    uicontrol('Style','pushbutton','String','Load Audio',...
        'Position',[20 20 100 30],'Callback',@loadAudio);

    function loadAudio(~,~)
        [file, path] = uigetfile('*.wav');
        if file ~= 0
            [audio, Fs] = audioread(fullfile(path,file));
            t = (0:length(audio)-1)/Fs;
            plot(ax1, t, audio); title(ax1, 'Signal in Time');
            N = length(audio);
            f = Fs*(0:(N/2))/N;
            Y = fft(audio);
            plot(ax2, f, abs(Y(1:N/2+1))); title(ax2, 'Spectrum');
        end
    end
end
```

```
    title(axes_processed, 'Processed Image');

    % Buttons for operations
    uicontrol('Parent', fig, 'Style', 'pushbutton', ...
            'String', 'Load Image', ...
            'Position', [50, 200, 120, 30], ...
            'Callback', @loadImage);

    uicontrol('Parent', fig, 'Style', 'pushbutton', ...
            'String', 'Grayscale', ...
            'Position', [50, 150, 120, 30], ...
            'Callback', @convertGrayscale);

    uicontrol('Parent', fig, 'Style', 'pushbutton', ...
            'String', 'Edge Detection', ...
            'Position', [50, 100, 120, 30], ...
            'Callback', @edgeDetection);

    uicontrol('Parent', fig, 'Style', 'pushbutton', ...
            'String', 'Equalize Histogram', ...
            'Position', [180, 150, 120, 30], ...
            'Callback', @equalizeHistogram);

    uicontrol('Parent', fig, 'Style', 'pushbutton', ...
            'String', 'Median Filter', ...
            'Position', [180, 100, 120, 30], ...
            'Callback', @medianFilter);

    % Variables to store images
    original_image = [];
    processed_image = [];

    % Callbacks
    function loadImage(~, ~)
        [filename, pathname] = uigetfile({'*.jpg;*.png;*.bmp',
            'Images'});
        if filename ~= 0
            original_image = imread(fullfile(pathname, filename));
            axes(axes_original);
            imshow(original_image);
            title(axes_original, 'Original Image');

```

8. Generate a vector `x` with 200 elements using the function `rand`. Extract to another vector all elements of `x` greater than the mean. Use the `for` and `if` statements.

9. Solve the previous exercise using the `while` and `if` statements.

10. Calculate the nodal admittance matrix of the network in Figure 15.1.

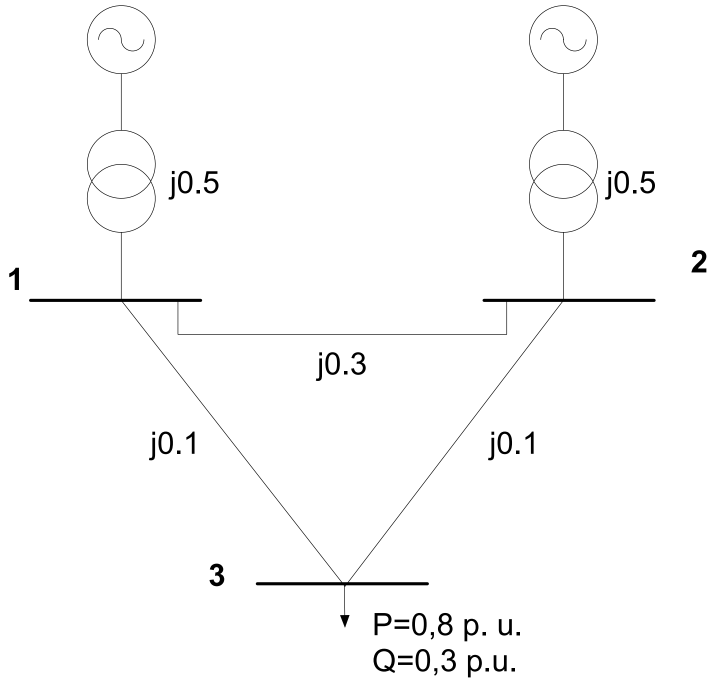


Figure 15.1: Network

11. Calculate the power flow in each line of the network in Figure 15.1 using the DC Model. Consider that:

$$\hat{\theta} = B^{-1} \cdot \hat{P} \quad \text{and} \quad P_{ik} = \frac{\theta_i - \theta_k}{x_{ik}}.$$

12. Write a MATLAB script that constructs the graph of the polynomial $p(x) = x^5 - 2x^4 + 6x^3 - 10x^2 + 15$ with a solid blue line, indicating the

MATLAB: FROM FUNDAMENTALS TO ARTIFICIAL INTELLIGENCE

FILIPE AZEVEDO

About the book

This book is an essential and comprehensive guide for students, professionals, and enthusiasts who want to explore and master MATLAB, a powerful platform for numerical computation, graphical visualization, and computational development.

Starting with the fundamentals of the language, the reader is led through a solid pedagogical progression that includes flow control, matrix manipulation, 2D and 3D graphics, scripts and functions, culminating in advanced applications in optimization, artificial intelligence, neural networks, deep learning, reinforcement learning, and modeling with Simulink.

With a practical and accessible approach, reinforced by real examples and progressive exercises, this work offers an effective learning curve, preparing the reader to face academic and professional challenges in the MATLAB universe.

About the author

Filipe Azevedo is an Associate Professor at ISEP, Polytechnic of Porto, and an integrated researcher at INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. He holds a Bachelor's and Master's degree in Electrical and Computer Engineering from the Faculty of Engineering of the University of Porto, and received his PhD in Electrical and Computer Engineering from the University of Trás-os-Montes e Alto Douro with distinction and honors.

He began his professional career at EFACEC Energia in the technical-commercial area, before dedicating himself to teaching and research in higher education. Since 2000, he has been teaching at ISEP, where he has been developing work in the areas of energy systems, artificial intelligence, and engineering economics.

His research at INESC TEC focuses mainly on modeling and optimization of smart electrical networks, electricity markets, integration of renewable sources, and the application of decision support techniques. His interests also extend to energy security, cryptography, and cybersecurity, with particular attention to the impact of these technologies in strategic contexts.

With more than two decades of academic and scientific experience, he has sought to articulate pedagogical practice with applied research, developing innovative solutions and training engineers with critical thinking and technological vision.

ISBN: 978-989-917-787-1

9 789899 177871

www.**engebook**.pt

engebook