–Etienne A. Mbe Mbock –

Algorithms in Matlab

Computer Engineering for Physics(TIP)

Ruprecht-Karls-Universität Heidelberg

January 10, 2013

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Mbe Mbock, Etienne Aubin:
Algorithms in Matlab - The Reality of Abstraction and the Power of Pseudocodes
ISBN 978-3-86376-033-5

All Rights Reserved
1. Edition 2012, Göttingen
© Optimus Verlag
URL: www.optimus-verlag.de

Printed in Germany
Paper is FSC certified (wood-free, chlorine free and acid-free,
and resistant to aging ANSI 3948 and ISO 9706)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, scanning, or otherwise without the prior written permission of the Publisher. Request to the Publisher for permission should be addressed to info@optimus-verlag.de.

Contents

1 Introduction **4**

2 How to write Algorithms **5**

2.1 Finding the Maximum of Three Numbers 5

2.2 Finding the Largest Element in a Finite Sequence 6

2.3 Finding the Largest Element in a Finite Sequence 7

2.4 Testing Whether a Positive Integer is Prime 8

2.5 Algorithm that finds the smallest Prime that exceeds a Positive given Integer n 9

2.6 Algorithm that finds an Item in a given Sequence of Numbers 10

2.7 Euclidean Algorithm (Largest Common Divisor) 11

2.8 Euclidian Algorithmus (Lowest Common Multiple) 12

2.9 Euclidean Algorithm (Recursive Largest Common Divisor Algorithm) 13

2.10 Euclidean Algorithm (Recursive Lowest Common Multiple algorithm) 14

2.11 p-Combinations of 1, 2, 3, ...n 15

2.12 Permutations of 1, 2, 3,...,n 16

2.13 Largest Item and Smallest Item of a given Sequence s 18

2.14 Finding the Location of an Item in a given Sequence s 20

2.15 Converting a Decimal Interger to Base b 21

2.16 Converting a Decimal Interger to Base b 22

2.17 Selection sort. The Algorithm selects the largest item 23

2.18 Non-recursive Selection of Sort Algorithm 24

2.19 Binary Search for an Increasing Sequence 25

2.20 Merging two increasing Sequences into a single sorted sequence 26

2.21 Merging two increasing Sequences into a single sorted sequence 28

2.22 Summary and Key Terms 30

3 Algorithms in Linear Algebra **31**

3.1 The Gaussian with Backward Substitution and Determinant Method 31

3.2 Modified Gauss Elimination for Determinant and Inverse Computation 35

3.3 Band Matrices-Tridiagonal Matrices 38
3.4 Pivoting Strategy and LU-Decomposition Without Row Interchanges 43
3.5 Lower-Diagonal-Upper and Cholesky Factorization 49
3.6 Lower-Diagonal-Upper Matrix Factorization 49
3.7 Cholesky Factorization 52
3.8 Summary and Key Terms 53
3.9 Jacobi-Method 53
3.10 Jacobi Iterative Procedure 54
3.11 Gauss Seidel Iterative Procedure 56
3.12 The SOR Iterative Procedure 59
3.13 Iterative Refinement 61
3.14 Summary and Keywords 62

4 Linear Programming **63**

4.1 Definitions and Notations 63
4.2 The ith-coordinate vector e in the m-dimensional Euclidean space 63

5 Linear Programming Mathematics **67**

5.1 Elements of Linear Programming 68
5.2 Homogeneous Systems of Linear Equations 68
5.3 Basic Solutions 69
5.4 Example 69
5.5 Example 79

6 Improving a Basic Feasible Solution **93**

7 General Representation of a Linear Programming Problem **107**

7.1 Summary and Key Terms 111

8 Conclusion **113**

9 References **115**

Preface

This book is written for mathematicians learning algorithms. The Algorithms that will be presented in this book are simple and do not require any prerequisites in mathematics. For those of you studying computer sciences, this book will provide you with the code of all elementary algorithms encountered in this book. The style we adopt in this book is simple. The presentation of the algorithms in pseudocode will not assume any background in mathematics or programming. This book will contain 7 chapters and the chapters focus will be the presentation of the algorithms that are related to the chapters. The first chapter is the introduction. The second chapter will contain algorithms in general. The third chapter presents algorithms of linear algebra. The fourth, fifth and sixth chapter will treat linear programming and linear programming mathematics. The seventh chapter presents the general representation of a linear programming. In chapter 8 we present the conclusion of the book.

1 Introduction

An algorithm is a set of instruction with precise characteritics. The characteristics of an algorithm are

- Precision
- Finiteness
- Uniqueness
- Generality
- Input-Output

Definition 1.1. *We understand under precision the possibility of an algorithm to be written with precised steps*

Definition 1.2. *The uniqueness occurs when the intermediate results of each step of execution are uniquely defined and depend on the inputs and the results of the previous steps.*

Definition 1.3. *We want to describe every algorithm as a finite characteristic. Finiteness means the algorithm will stop after finitely many instructions have been executed.*

Definition 1.4. *An algorithm will receive an input and produce an output*

Definition 1.5. *The algorithm will be applied to a set of input*

2 How to write Algorithms

The best way to describe an algorithm is to use the ordinary language. But many computer scientists and mathematicians prefer pseudocode because of its universality and structure. The pseudocode we write will resemble the Matlab language. Next to the pseudocode will be given the matlab version of the algorithm.

2.1 Finding the Maximum of Three Numbers

Algorithm 2.1.

function MAXIMUMTHREENUMBER(a, b, c)
  $x \leftarrow a$
  if $b > x$ ***then***
    $x \leftarrow b$
  end if
  if $c > x$ ***then***
    $x \leftarrow c$
  end if
  return x
end function

Listing 1: Algorithm Maximum of Three Numbers

```
1  function x=MaximumThreeNumber(a,b,c)
2                 x = a;
3                 if ( b > x )
4                 x = b;
5                 end
6                  if( c > x )
7                  x = c;
8                  end
9                  end
```

This is a quite simple algorithm for determining the maximum of three numbers using the only if then structure. We assume that the maximum of the three numubers is x. We then start with the first number a with the if structure we update the value of x. At line 9 the value of the maximum of the 3 numbers is given and completes the algorithm.

2.2 Finding the Largest Element in a Finite Sequence

We can use the while loop structure to determine the largest number in a sequence S

Algorithm 2.2.

function FINDLARGESTELEMENT*(*S*)*
  $large \leftarrow s_1$
  $i \leftarrow 2$
  while $i \leq n$ ***do***
    if $s_i > large$ ***then***
      $large \leftarrow s_i$
    end if
    $i \leftarrow i + 1$
  end while
  return $large$
end function

Listing 2: Finding the Largest Element in a Finite Sequence

```
1   function FindLargestElement(s)
2            n = length(s);
3           large = s(1);
4           i = 2;
5           while (i <= n)
6           if s(i) > large
7           large = s(i);
8           end
9           i = i + 1;
10          end
11          disp(large);
12          end
```

2.3 Finding the Largest Element in a Finite Sequence

Algorithm 2.3.

function FINDLARGESTELEMENT*(S)*
  $n \leftarrow length(S)$
  $large \leftarrow s_1$
  for $i = 2, ..., n$ ***do***
    if $s_i > large$ ***then***
      $large \leftarrow s_i$
    end if
  end for
  return $large$
end function

Listing 3: Finding the Largest Element in a Finite Sequence

```
1  function FindLargest-Element(s)
2           n = length(s);
3           large = s(1);
4           for i = 2:n
5           if s(i) > large
6          large = s(i);
7          end
8          end
9          disp(large);
10         end
```