

Advances in Science, Technology & Innovation  
IEREK Interdisciplinary Series for Sustainable Development

Zekâi Şen

# Earth Systems Data Processing and Visualization Using MATLAB



@Seismicisolation



---

# **Advances in Science, Technology & Innovation**

## **IEREK Interdisciplinary Series for Sustainable Development**

### **Editorial Board Members**

Anna Laura Pisello, Department of Engineering, University of Perugia, Italy  
Dean Hawkes, Cardiff University, UK  
Hocine Bougdah, University for the Creative Arts, Farnham, UK  
Federica Rosso, Sapienza University of Rome, Rome, Italy  
Hassan Abdalla, University of East London, London, UK  
Sofia-Natalia Boemi, Aristotle University of Thessaloniki, Greece  
Nabil Mohareb, Beirut Arab University, Beirut, Lebanon  
Saleh Mesbah Elkaffas, Arab Academy for Science, Technology, Egypt  
Emmanuel Bozonnet, University of la Rochelle, La Rochelle, France  
Gloria Pignatta, University of Perugia, Italy  
Yasser Mahgoub, Qatar University, Qatar  
Luciano De Bonis, University of Molise, Italy  
Stella Kostopoulou, Regional and Tourism Development, University of Thessaloniki, Thessaloniki, Greece  
Biswajeet Pradhan, Faculty of Engineering and IT, University of Technology Sydney, Sydney, Australia  
Md. Abdul Mannan, Universiti Malaysia Sarawak, Malaysia  
Chaham Alalouch, Sultan Qaboos University, Muscat, Oman  
Iman O. Gawad, Helwan University, Egypt

### **Series Editor**

Mourad Amer, Enrichment and Knowledge Exchange, International Experts for Research, Cairo, Egypt

**Advances in Science, Technology & Innovation (ASTI)** is a series of peer-reviewed books based on the best studies on emerging research that redefines existing disciplinary boundaries in science, technology and innovation (STI) in order to develop integrated concepts for sustainable development. The series is mainly based on the best research papers from various IEREK and other international conferences, and is intended to promote the creation and development of viable solutions for a sustainable future and a positive societal transformation with the help of integrated and innovative science-based approaches. Offering interdisciplinary coverage, the series presents innovative approaches and highlights how they can best support both the economic and sustainable development for the welfare of all societies. In particular, the series includes conceptual and empirical contributions from different interrelated fields of science, technology and innovation that focus on providing practical solutions to ensure food, water and energy security. It also presents new case studies offering concrete examples of how to resolve sustainable urbanization and environmental issues. The series is addressed to professionals in research and teaching, consultancies and industry, and government and international organizations. Published in collaboration with IEREK, the ASTI series will acquaint readers with essential new studies in STI for sustainable development.

More information about this series at <http://www.springer.com/series/15883>

---

Zekâi Şen

# Earth Systems Data Processing and Visualization Using MATLAB



@Seismicisolation

Zekâi Şen  
Department of Civil Engineering, Faculty of  
Engineering and Natural Sciences  
Istanbul Medipol University  
Beykoz, Istanbul, Turkey

ISSN 2522-8714                   ISSN 2522-8722 (electronic)  
Advances in Science, Technology & Innovation  
IEREK Interdisciplinary Series for Sustainable Development  
ISBN 978-3-030-01541-1       ISBN 978-3-030-01542-8 (eBook)  
<https://doi.org/10.1007/978-3-030-01542-8>

Library of Congress Control Number: 2019932682

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland



**RAHMAN VE RAHİM OLAN ALLAH'IN ADI İLE**

**IN THE NAME OF ALLAH, THE MOST MERCIFUL, THE MOST  
COMPASSIONATE**

*I hope that each individual will try to perform his/her intellectual ability and moral behavior along an increasing trend for the common share and prosperity of humanity in homogeneous and isotropic manner*

---

## Preface

Developments in science and technology are not possible prior to knowing what the philosophical and then especially logical foundations of any phenomenal investigation or problem solutions are based first on linguistical information, which are then translated to symbolic logic with crisp logic principles in the mathematical equation, and formulation forms. Arrival to any mathematical equation systems after logical rules, there are two ways to benefit from these expressions as either to try and to solve the problem under the guidance of arithmetical and mathematical principles, which was the main road about half a century ago, but today the mathematical equation systems or algorithms can be solved speedily in a short time duration by digital computer aids provided that a program is written in some computer language. In many education systems all over the world rather than science philosophical based logical derivations of many formulations, their presentations and explanations are provided either in the form of spoon-feeding or mechanically through memorization and without actual dynamic brain activities. Especially in recent decades, there are plenty of software in any domain desired and they become an arena of commercialization, and therefore, rather than trying and writing simple service satisfactory computer programs they are bought in wholesale and without knowing the internal logical structure the guidance of sequenced button pushes and clicks on computer screen lead to desired solutions. It is emphasized that to be able to write one's own computer programs for his/her research the prime condition is to get oneself familiarize with the logical bases of the problem at focus or write down the logical rule base so that they can be converted to a computer language such as the MATLAB, which is the sole presentation in this book. It is advised that whoever wishes to write his/her own program the first step is to put down into logical premises the problem at hand verbally, because each premise corresponds to a computer language statement or command.

There are different disciplines covered in this book within the earth sciences systems domain including atmospheric research, meteorology, hydrology and water science, water quality variations, groundwater resources assessment, various simulation works concerning the topics covered in various chapters, climate change impact and downscaling procedures, spatial analysis and modeling, renewable energy calculation methodologies, and general tendency determination in any records especially in the form of time series. In each chapter, various computer programs in MATLAB language are presented with applications. Each program is self-explanatory in open source form and can be modified accordingly.

The content of this book is based on the vast experience of the author, especially in the arid region of the Arabian Peninsula through his academic work at the King Abdulaziz University, Faculty of Earth Sciences, Kingdom of Saudi Arabia; at the application establishment of the Saudi Geological Survey, Jeddah; and also at the Meteorology and Civil Engineering Faculties at the Istanbul Technical University, Istanbul, Turkey, and Turkish Water Foundation.

I hope that this book will support to those interested in flood discharge estimation with risk attachments, climate change relationships, hazard and mitigation aspects, and their applications in flood prevention works. I thank my colleagues, who have encouraged me to write a

book on floods, and especially my wife Mrs. Fatma Şen, who had kept silence, endurance, and patience during my extensive hourly, daily, monthly, and yearly works for the preparation of this book.

Kavacık, Istanbul, Turkey  
August 2018

Zekâî Şen

---

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	General	1
1.2	Science Philosophy	2
1.3	Software and Philosophy	3
1.4	Logic Rules	4
1.5	Computation	5
1.6	Computer Program Structure	5
1.7	Purpose of the Book	6
	References	6
<b>2</b>	<b>Meteorology</b>	7
2.1	General	7
2.2	Storm Rainfall Records	7
2.2.1	Frequency (Risk) Calculation from Probability Distribution Function (PDF)	9
2.3	Intensity–Duration–Frequency (IDF)	12
2.3.1	Dimensionless Intensity–Duration (DID) Curve	15
2.3.2	PDF and IDF Curve Software	17
2.4	Radar Data	21
2.4.1	PDFs Matching Procedure	23
2.4.2	Application	28
2.5	Probable Maximum Precipitation (PMP)	30
2.6	Meteorological Water Balance	36
2.7	Standard Precipitation Index (SPI)	43
2.8	Harmonic Analysis	46
2.8.1	Known Seasonality Case	50
	References	52
<b>3</b>	<b>Hydrology</b>	53
3.1	General	53
3.2	Digital Elevation Model (DEM)	53
3.2.1	Gradients (Slopes)	54
3.2.2	Cross-Sections	54
3.2.3	Rating Curve	55
3.3	Floods	67
3.3.1	Flood Risk	72
3.3.2	Probable Maximum Flood (PMF)	73
3.4	Risk Calculation	74
3.5	Droughts	78
3.5.1	Wet and Dry Periods	78
3.5.2	Drought Indices	80
	References	87

<b>4</b>	<b>Hydrogeology</b>	89
4.1	General	89
4.2	Aquifer Parameters	89
4.3	Confined and Unconfined Aquifer Tests	90
4.3.1	Single Porous Media	91
4.4	Fractured Aquifer Test	106
4.5	Groundwater Management and Optimization	110
4.5.1	Well Field Optimization	113
	References	114
<b>5</b>	<b>Hydrochemistry</b>	115
5.1	General	115
5.2	Ion Concentration	115
5.2.1	Ion Conversion Units and Balance	115
5.3	Numerical Indicators	117
5.3.1	Electrical Conductivity (EC)	117
5.3.2	Total Dissolved Solids (TDS)	118
5.3.3	Total Hardness	118
5.3.4	Sodium Adsorption Ratio (SAR)	119
5.4	Graphical Representations	119
5.4.1	Stiff Diagram	119
5.4.2	Circular Diagram	120
5.4.3	Schoeller Diagram	120
5.5	Trilinear (Piper) Diagram	121
5.6	Multi-rectangular Diagram	126
5.7	Standard Ion Index Template	130
5.8	Water Quality Index	133
5.8.1	Similarity Model	134
5.8.2	Association Matrix	136
5.8.3	Fuzzy Classification	137
5.9	Applications	137
	References	140
<b>6</b>	<b>Simulations</b>	143
6.1	General	143
6.2	Stochastic Processes	143
6.2.1	Independent Process	145
6.2.2	Markov Process	146
6.2.3	White Markov Process	148
6.2.4	ARIMA Process	150
6.3	Maximum Rainfall Simulation	155
6.4	Computer Programs	163
	Appendix	164
	Seasonal Generation Process	164
	References	170
<b>7</b>	<b>Climate Change</b>	171
7.1	General	171
7.2	General Circulation Models (GCMs)	171
7.3	Fundamentals of Climate Modeling	172
7.3.1	Basic Flowchart	172
7.4	Downscaling Principles	174
7.4.1	Dynamic Downscaling Procedure	174
7.4.2	Statistical Downscaling Procedure	174

---

7.5	Simple Statistical Downscaling Model . . . . .	175
7.5.1	Spatial Dependence Function . . . . .	176
7.5.2	Spatial Model . . . . .	177
7.5.3	Temporal Model . . . . .	179
7.6	Software for Statistical Downscaling . . . . .	180
7.6.1	Application in Turkey . . . . .	187
	References . . . . .	190
<b>8</b>	<b>Spatial Analyses . . . . .</b>	<b>193</b>
8.1	General . . . . .	193
8.2	Spatial Dependence Function (SDF) . . . . .	193
8.3	Semi-variogram (SV) . . . . .	195
8.4	Cumulative SV . . . . .	197
8.5	Point CSV . . . . .	200
8.6	Directional Semi-variogram (DSV) . . . . .	202
8.7	Innovative Spatial Dependence Function . . . . .	207
8.8	Regional Estimation Procedure . . . . .	210
	References . . . . .	215
<b>9</b>	<b>Renewable Energy . . . . .</b>	<b>217</b>
9.1	General . . . . .	217
9.2	Solar Energy . . . . .	217
9.2.1	Angström Model . . . . .	217
9.2.2	Unrestrictive Model . . . . .	219
9.2.3	Successive Substitution Model . . . . .	221
9.2.4	Nonlinear Models . . . . .	223
9.2.5	Probabilistic Innovative Solar Energy Estimation . . . . .	225
9.3	Wind Energy . . . . .	229
9.3.1	Topographic Features . . . . .	232
9.3.2	Power Formulations . . . . .	232
9.3.3	Wind Field Data . . . . .	233
9.3.4	Turbine Calculations . . . . .	234
9.4	Hydropower . . . . .	239
9.4.1	Classical and Energy Tree Hydropower Models . . . . .	239
9.5	Newtonian Energy Principle . . . . .	240
	References . . . . .	242
<b>10</b>	<b>Innovative Trends . . . . .</b>	<b>243</b>
10.1	General . . . . .	243
10.2	Trend Concepts . . . . .	243
10.3	Innovative Trend Template (ITT) Analysis . . . . .	245
10.4	Innovative Variability Analysis . . . . .	247
10.5	Crossing Trend Identification . . . . .	249
10.6	Variation Trend . . . . .	259
10.7	Partial Trend Components . . . . .	261
10.8	Fuzzy Trend Analysis . . . . .	262
10.9	Over-whitening Trend Analysis . . . . .	267
	References . . . . .	274
	<b>Index . . . . .</b>	<b>275</b>



# Introduction

1

## 1.1 General

In recent years, there is a tremendous increase in the interaction between art, science, and technology through the unprecedented developments in the computer technology. Almost all aspects of life are directed by computer software in an increasing number of software, application extent, size, and variety of topics. The number of computer science departments in almost all the universities is more than enough, because computer experts in hardware and especially software are bound to find job positions rather easily in many government offices and private companies. Different specialists from a variety of disciplines are coming together in a productive team for better software development in the service of daily life activities. All over the world there are collaborating interdisciplinary activities around the computation and software development.

Computer software engineering career has numerical which has numerical application directions based on philosophical (linguistical fundamentals of the problem) and logical interpretations of available equations and formulations. Classical engineers that do not have artistic thoughts and background cannot be fruitful in software program writing. In many engineering education institutions all over the world, unfortunately, the artistic facet of engineering is forgotten and verbal ingredients are ignored completely as if engineering does not need philosophy and logical aspects for problem-solving. Engineers are trained for numerical calculations only even without the consciousness of the basic root philosophical and logical bases of equations and formulations, which are in the form of symbolic logic forms. Without these constituents, it is not possible to make inferences through dynamic rational thinking. Even the experience gained throughout the years cannot be productive without them. Knowledge generation and innovative productions are possible with philosophical and logical twins

only; otherwise one cannot provide a functional mechanism of any methodology in detail with creative thought that may root future developments. It is, therefore, necessary to depend on available methodologies and techniques with critical reasoning and questioning. Especially, software engineering depends on these features, which are indispensables in software writing for execution on digital computers. Experience has shown that the more one writes software programs the more he/she becomes acquainted with the logical reasoning even though it may be approximate at different phases.

Philosophical and logical principles provide a more effective and creative mind activity in one's thoughts leading to new ideas, formulations, methodologies, procedures and even to the modification of the existing information, equation, or algorithm.

Critical thought and its fruitful consequences can be obtained by philosophy learning and teaching in order to be empowered by critical view abilities that can revise available information with rational, philosophical, and logical rules. In this way, information boils down in his/her memory linguistically for any future verbal, and symbolic formulation generations. This is very important for those who would like to be a demanded software program writer. Critical thoughts are based on reading, writing, and productions in alive. Remnants of memorization settle down even after repetitive dependence on philosophy and logic. Hence, in certain affairs one becomes rather automatic in laying down the logical steps for the problem at hand, and once the logical rules ripen in the mind then its translation to computer language completes the task. Provided that philosophical knowledge filters through the logical principles then one can provide human or computer linguistic translation without difficulty. This gives to the individual the ability to feel that he/she can also generate information and can use knowledge at proper times and locations for problem solution through the computer software written by him (her) self.

## 1.2 Science Philosophy

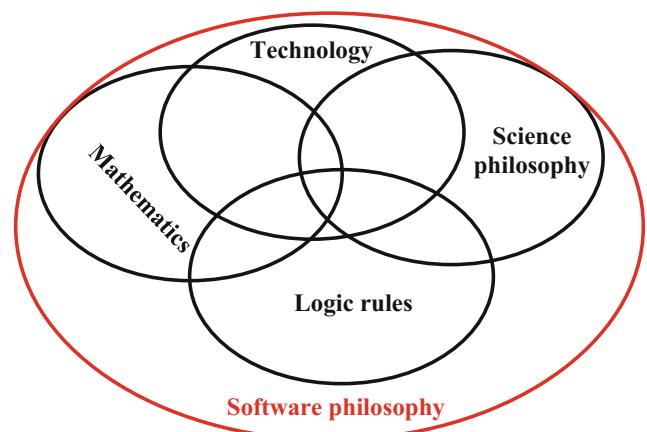
Philosophy of science has an active role in the scientific studies since the science became rather independently spelled out from the philosophy after the renaissance period. During the last decade through patent institutions, philosophy of technology started to ripen and there are many articles in the literature about such aspects (Scharff and Dusek 2003; Meijers 2009). However, philosophy of engineering is a very recent debate in the world, since the last several years (De Poel and Goldberg 2010; Sen 2014). Hence, philosophy of science is an overlooked or delayed aspect of systematic engineering creative thinking. This does not mean that engineers never benefit from the philosophical principles, of course they do, but this remained on individual bases and it could not be systematized during the engineering education. For instance, ethical behaviors and career rules are based on philosophical basis, but real mass of engineering background on tangible aspects are far away from systematic philosophical principles. Even the entrance of ethics into the engineering domain is a work achieved during the last several decades (Davis 1998, 2005, 2012). One can easily say that philosophy of engineering is almost nonexistent in engineering curricula. Another vague entrance of philosophical thinking into engineering domain may be that generally, engineering and technology are thought as distinct disciplines, but there are occasional interferences between the two, which transfer some philosophical aspects into engineering thinking, because some engineers take an active involvement in technological developments such as the first atom bomb production. Some may propose that there is no need for separate philosophy principles in engineering, because they are included in the philosophy of technology. This is not acceptable completely due to the benefit of engineers from the end products of science rather than technology in their mental formation outputs such as design, solutions to many problems, and in report writing. Most of the engineering students in the world cannot write proper reports or articles due to their inefficient philosophical backgrounds (Sen 2014). Many researchers may not sense that the problems are full of philosophical arguments. They try to settle down the problems by a set of rational, empirical, and logical rules, which constitute the basis of science philosophy.

The basis of any software passes through philosophical thinking for the logical, rational, verbal, and analytical steps to reach valuable equations to communicate with digital computers. It is not possible to generate any software without considering these essential and fundamental steps. Unfortunately, in many universities and research centers rather than concentration on philosophical solution of problem concerned, many researchers depend mostly on

readily available software. Science philosophy principles and contents are intact of various environment, energy, economy, sociology, engineering disciplines, etc. As a result, generative and innovative thinking is missing in research and development. Especially, in developing countries and societies ready programs are in use provided that data base is ready without any rule base consideration.

One cannot have the ability or even courage to sit down and write software even in his/her domain of activity if philosophy of science and its successor logical principles are not known and applied properly. Philosophy of science and logic provide approximate reasoning about the phenomenon mechanism and its internal function of operations. Technological innovative inventions are based on experience and expert view under the light of philosophical thinking. It is well recognized by many that science is concerned with discovery, technology with invention whereas engineering is more craft work concerned with making, producing, and generating alternative, simple, fast, effective, practical, and economical solutions for a given problem. Some may propose that there is no need for separate philosophy principles in engineering, because they are included in the philosophy of technology.

Figure 1.1 represents a mixture of different disciplines for a search of possible relationship in different proportions between mathematics, science, philosophy, logic, and technology. The effective combination of these disciplines helps to trigger the initial basis for the problem solution, which can be further studied by time for better solutions. Critical scientific knowledge including theories falls within the domain of science with continuous development throughout centuries. Patterns and blueprints are invention imprints in technological developments. However, engineering tasks are concerned with material products and designs.



**Fig. 1.1** Interaction among philosophy, logic, science, and technology

### 1.3 Software and Philosophy

Software is necessary for the speedy calculation of complicated functional performances that are based on mathematical equations. The software is the only way to instruct computers to do what is required through programs that are written in different computer languages. Successful software writing is possible through the science philosophical principles and logical rules in the form of mathematical expressions that are translated to computer programming statements. The experts that are capable to write software well understand the basic concepts in details of the logical roots of the problem at hand. The only way of writing software is through the philosophical and logical understanding of the problem confronted. It is not necessary that the computer programmer should know all the necessary concepts, but anytime he/she is requested to write software at the service of the client, then the client must explain to him/her all the detailed philosophical and logical rules and ingredients.

Software and the philosophy are interrelated through the design, because the software itself is a design and the philosophy explains the design. Philosophy dictates software and software request dictation from the philosophy of the problem. The reader must not be discouraged by word "philosophy," because it is not the formal philosophy that one should know, but software philosophy, which is the linguistic (verbal) explanation of the problem-solving mechanism by logical statements in the form of predicates and related consequences. Software design trains also the programmer continuously until a satisfactory solution is reached. In practice, it is not possible to arrive at success in the first trial of the software program, but after a series of logical and syntax errors through trial-and-error procedure the final goal can be achieved. A software writer does not need to indulge in formal philosophical principles and not even the history of the philosophy. He/she has to keep in mind that the philosophy in software writing is equivalent to verbal knowledge and information about the problem concerned. In the meantime, the mathematical formulations are also important. It is well known that mathematics foundation is logic and the software writing depends also on mathematics and logic.

After all what have been explained above one can feel that the philosophy of the problem concerned influences directly the software. In general, philosophy is abstract, but software is based on scientific principles, and therefore, one should be concerned with the philosophy of science only. However, by the time the software also becomes abstract, because its core may not be understood by everyone equally, and the sole person for its command is the software programmer. In daily life, the software becomes even more

abstract, because most of the users are not aware of the internal mechanism of the software, but only input and output ends are important for those who are making use of any software for systematic, accurate, speedy, and quick end results. Rhetoric is a significant part of the philosophy for software writing. In the explanation of a ready software rather than crisp concepts verbal explanations are provided and these lead to mutual discussion, which may be felt as a philosophical debate especially for those who may not be expert or experienced in the subject of the problem. To grasp software a harmonious mixture of intellectual thinking, electronic learning, and system design stages are necessary.

Software is not tangible like hardware, and it depends on the flowchart of rational steps in algorithmic solutions of mathematical functions. It is necessary for computer instruction through a set of statements within a program that starts execution leading from given informative data to conclusive numerical outputs. Software can be written provided that the rational philosophical background is known by means of logical steps. Either software programmer should be acquainted with the philosophical and logical generation mechanism of the problem concerned or he/she will be informed by a client according to the desired problem solution. The conceptual principles of the problem at hand will be transferred to the programmer, who will set down the statements from the logical rules of the problem. It is advised that software should be checked with well-known or simple solutions prior to their effective and general uses. Software program design is intimately related first to science philosophy and subsequently to logical principles and rules. Philosophy depends on abstract prepositions, but software is a scientific concept and design. Generation of software for any problem is an art, which helps to cure the scientific problems of others in short time durations. It is also a commercial material that brings income to many individuals and companies. Although philosophers might not be interested in software writing, software programmers can achieve their goals after the establishment of science philosophical principles. Today, engineering, economic, social, and any sort of human activity cannot be thought without software running in the computers. Software needs thinking and reasoning even though it may be approximate. At times software writing depends on the mental thinking with the generation of ideas from the soul, and hence, it may be subjective, but the product is always objective. One should consider the verbal statement "Garbage in garbage out," which implies that one cannot reach to plausible output if the reasoning data are not rational and correct. Although in many education systems there are compulsory courses for computer programming leading to software, it seems that not everybody can benefit from such training, because software writing is rather personal wonder, ambition, and wish.

Almost all software requires numerical data in the form of basic information according to crisp logic (zero-one, white-black); recently fuzzy logic principles have entered the domain that also digests verbal knowledge and information. Software are very effective almost on everyday life with increasing trend of data mining and processing in the best, cheap, optimum, and efficient manner.

Mathematical formulations are arsenal of logistic and linguistic elements that need for analytical, empirical, and numerical or simulation solutions. Unfortunately, worldwide in many education institutions due to missing of linguistic explanations and logical principles, the bases of mathematical formulations cannot be thought properly, and consequently, the domain is left with memorization without critical revision and rational or even approximate reasoning. Additionally, lack of science philosophical reasoning and suspicion from scientific information pave the way to mechanical prescriptions for desired solutions. Even the software that needs at least logical steps is undertaken as curing pills for scientific advancement without fruitful results. It is a common illness in scientific thought that although rationality is spelled out frequently with its void understanding and application, it is taken for granted that scientific knowledge and information are rational or have rational basis. Hence, many rely on the false belief that science is faith that does not need any criticism. Most often thought constitutes classical educational system structure. It should be taken into consideration that for scientific development each word or terminology has etymologic root and epistemological background meanings. In order to surface these meanings, the main keys lie in the questions of how? and why? Without answers to these two questions, the knowledge cannot be transmuted to scientific and generative information. Answers to these questions lead to the fundamentals of scientific knowledge. Rational thinking steps are imagination of the objects, their visualization in the form of designs, i.e., geometry (figures, shapes, etc.), and finally information production. The mind is capable to perceive not only Euclidian geometry as points, lines, volumes, but overridingly shapes of any object in general.

## 1.4 Logic Rules

In computer science, hardware design is based on physical and material sciences, whereas the software design is dependent on logic supported by linguistically and mathematical expressions. Linguistic thoughts, interpretations, evaluations, and assessments are products of the philosophy about phenomenon or problem concerned, and they form the structure of the software. In logical foundations, ideas, concepts, system, and priority among the most simplistic expressions are needed for successful programming.

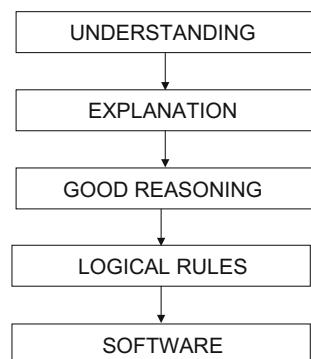
The basis of relationship search is to try and relate causative (inputs) variables to result (output) variables. The relationship search seeks for truth, belief, and justification, which are subjective, but common sense and logical principles provide an objective solution.

Behind any equation, formulation, or numerical solution, there is linguistic arena that led to mathematical formulations, and hence, software writing may become rather easy. It is not possible to write computer program with missing linguistic explanation without science philosophy. It is asserted that all the scientific knowledge is rational. Imagination is necessary for visualization and computation. Rationality provides mathematical relationships through logical rules. Based on the logical thoughts, empiricism is essential to make sense of the world. Subjective thinking penetrates objectivity domain by time through imagination and visualization, and hence, there is not a crisp boundary between subjectivity and objectivity. Empirical works which are based on either observations or measurements as experimental information help to decrease the degree of subjectivity by means of increasing logical concepts. The scientific principles include vagueness, incompleteness, and uncertainty, and hence, they can be considered by fuzzy logic principles, which are then defuzzified into crisp forms of logical rules (Şen 2014). The fuzzy aspects are left outside the content of this book.

Rational inferences have a logical basis, but the final confirmation should be achieved through convenient tests experimentally. Approximate reasoning based on logical principles leads to a set of rules that can be translated to software language, which is MATLAB in this book. The sequential steps as in Fig. 1.2 guide to software writing.

Human wonder and mind serve for solving problems through philosophical (linguistic information) thinking, logical rules, and experience and expert views. The logical concepts in understanding complex problems are dependent on observations, experiences, and conscious expert views.

**Fig. 1.2** Software steps



## 1.5 Computation

Scientific computation is within the daily life activities in any discipline for organization, clustering, estimation, and prediction purposes. Almost all the governmental and especially company tasks end up with the necessary calculations, which require speedy and accurate computation facilities that are nowadays satisfied through the digital computer software. Computation capabilities are increasing day by day toward better understanding and solving complex problems automatically. The most important preliminary aspect in any computation is to have linguistic and numerical algorithms that guide one step by step from the beginning to the end of problem solution. The conversion of these steps into an automatic solution mechanism is possible through proper mathematical models, simulation techniques, and their computer software in social, economic, engineering, and other human activities.

Computational problems can be achieved apart from the algorithms with the development of optimization techniques through hardware, software, network, and especially, data mining and management works. These are the main means to solve demanding computational problems. Recently, information technology (IT) and scientific principles through the support of software engineers pave their final goal toward success.

In various scientific disciplines, the problem solution lies in the availability of numerical analysis computation and simulation techniques. Although there are field works especially in earth systems and laboratory works in many disciplines, their replacements or supports are taking place through the computational techniques by means of convenient software. Field and laboratory collected data are stored in the memory of computer, and they are treated with convenient methodological computations by means of software. In this manner, field and laboratory works remain as traditional research units, which cannot be without the support of software computation. Scientific computation methodologies help to gain and understand by means of mathematical model implications. Software is developed based on computer programs that model the system concerned with its input and output variables and data. Models may require massive computation procedures, and even for some problem-solving such as numerical weather prediction, global warming, and climate change, space research tasks need supercomputer network.

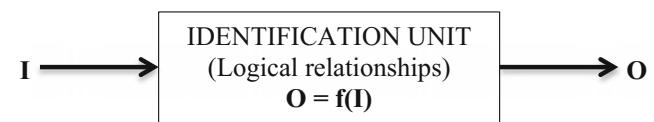
Especially, numerical computation and simulation aim various purposes in complex research problems. Rare events such as floods, earthquakes, and the like cannot be examined in the field or in the laboratory, but their past records provide a basis for numerical simulation and calculation opportunities. In this manner, future predictions can be obtained for

unobservable cases with certain risk levels. Computational tasks are also necessary for model fitting to available information and data after proper data analysis. Mathematical optimization and search procedures for the best solution among a set of alternatives also require computational affairs. For instance, there are different versions of numerical weather prediction model software and each one results in rather different prediction sequence for future. Hence, there is an ensemble of rational and logical computation solutions and the most convenient one can be obtained through an optimization criterion in digital computers.

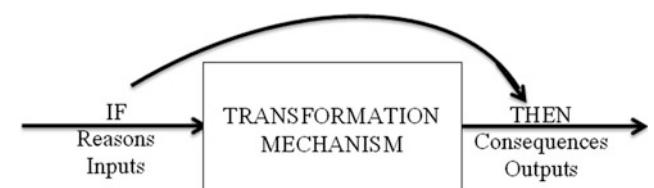
## 1.6 Computer Program Structure

Programming languages are basic means of conveying the abstract ideas, conceptions, definitions, logical relationships to a software program for the solution of the scientific and technological conceptual visualizations and imaginations in a virtual media. A computer software programmer is like a person, who can visualize the rational and especially logical relationships between different elements of a system input, I, and output, O, variables through an identification unit as in Fig. 1.3.

This figure provides a simple system modeling, which is in the interest of any mathematical modeler in science and technology provided that the functional relationships are identified scientifically. However, for a computer programmer apart from the symbolic equations their internal logical structures are the most important ingredients for converting them into software programming statements. It is, therefore, necessary to establish a logical rule base for computer program writing. Each statement must be in the form of logical statements, which are referred to predicates (Şen 2014), and their forms have two parts as preliminary (reasons) and consequent (result) (parts). The mathematical formulations must be abstracted in the form of following logical statements.



**Fig. 1.3** Mathematical program models



**Fig. 1.4** Computer program model



**Fig. 1.5** Software role

#### **IF (Antecedences) THEN (Consequent)**

or in terms of the simplest form as,

#### **IF (Reasons) THEN (Consequences)**

For a computer programmer, it is better to visualize the problem at hand not in the form of Fig. 1.3, but as in Fig. 1.4.

The preliminary step is a genuine logical and uncertain conceptualization of the phenomenon with its causal and resultant variables that are combined through a set of logical propositions in the form of software (Fig. 1.5).

The software is for the transformation mechanism, which translates the logical rules and mathematical expressions by a certain computer language, which is adapted in this book as the MATLAB.

## 1.7 Purpose of the Book

Science aspects that are concerned with various disciplines (engineering, earth systems, space research, and the like) need numerical computation procedures and algorithms of data collected from the field measurements or laboratory records. The same is also valid for data processing in social sciences and economics. Some of the data assessment and processing procedures are at large scales and complex and, therefore, require effective and efficient computer programs. Data reduction and graphical display in addition to probabilistic, statistical, stochastic, and simulation models and calculations are among the general purpose of the book. Not only students but also researchers in the universities and experts in different companies need to depend on reliable software. Especially, potential users of MATLAB in earth systems need a guidance book that covers a variety of practically applicable problem solution software.

The main goal of the book is to acquaint late undergraduate and graduate students, researchers and experts, engineers, environmentalists, earth scientists, hydrogeologists, hydrologists, meteorologists, and practicing experts concerned with the topics that are currently needed in different disciplines and earth systems scopes. In these

activities, it is essential to search for relevant literature, reports, and books for the solution to the problem at hand. This book includes the widest extent of the topics in the earth systems domain with brief, rationally understandable background information followed by the MATLAB software and factual data application examples. The supplementary electronic material is available online through the publisher's Web site for direct use of the readers.

For the software MATLAB is used for all the applications in each topic covered in the book especially including temporal time and spatial series. Among the main topics of the book are advanced statistical methods, because the usual methods are already available in the MATLAB programming library for ready use—stochastic processes, geostatistics and Kriging, digital elevation model applications, hydrogeological evaluation of aquifer tests, hydrochemistry, droughts, floods, climate change, rainfall analysis, risk management, and related aspects. The numerous examples help to demonstrate the usage of MATLAB in earth systems problem solutions. The reader is capable to modify all software to suit his/her own data set, because all the software are presented as open source.

## References

- Davis, M. (1998). Thinking like an engineer: studies in the ethics of a profession. New York/Oxford: Oxford University Press.
- Davis, M. (2005). Engineering ethics. Aldershot/Burlington, VT: Ashgate.
- Davis, M. (2012) "Ain't no one here but us social forces": Constructing the professional responsibility of engineers. *Science and Engineering Ethics* 18: 13–34.
- De Poel, I.V., Goldberg, D.E., (2010). Philosophy and engineering. An Emerging Agenda, Springer, Dordrecht, 361 pp ISBN 9789048128044.
- Meijers, A., ed. (2009). Philosophy of technology and engineering sciences (Handbook of the philosophy of science, volume 9). Amsterdam: North-Holland.
- Scharff, R. C., and V. Dusek, eds. (2003). Philosophy of technology: the technological condition. Malden, MA/Oxford: Blackwell.
- Sen, Z., (2014). Philosophy, Logic, and Scientific Perspectives in Engineering. Springer Verlag, Intelligence Systems Reference Library 143, 260 pp.

# Meteorology

# 2

## 2.1 General

The root meaning of word meteorology is the science of moving things in the air, which is referred to as the troposphere or the lower layer of atmosphere. Among such things are cloud formation and movement, wind and its speed, particulate matter and aerosol, precipitation (rainfall, snow and hail), temperature and pressure differences, water vapor, which are meteorological variables. There are theoretical modeling approaches for the tropospheric movement (general circulation models, GCMs) in large scales (Chap. 7). In practice, the meteorological variable variations are measured at a set of meteorology stations as time records, which help to identify temporal variations (trend, jump or shift, seasonality, and stochasticity) at the record site. Spatial and regional assessments are based on a set of temporal records leading to various meteorological and climate variation maps. Precipitation and temperature records are indispensable essentials in many engineering and societal works including water resources management, water allocation and distribution, droughts, floods, groundwater recharge (Chap. 3) and also in the climate change implication assessments (Chaps. 7 and 10).

In this chapter, the most significant two meteorological variable records (precipitation and temperature) are discussed from temporal perspectives with MATLAB programs. Furthermore, some fundamental calculation methodologies concerning the precipitation record evaluation are presented after brief explanations leading to MATLAB software programs.

## 2.2 Storm Rainfall Records

Precipitation is the amount of water that originates from the clouds under certain conditions and reaches the earth surface in rainfall (fluid) or snow and hail (solid) particle forms. It is necessary to have a set of standards for their measurements and mutual comparisons on an equal footing. In general, the

measurements provide discrete time records, and in particular, as a continuous record of a single storm rainfall in case of recording rain gauge availability (Sen 2017). The literature is full of rainfall time series evaluation based on the probabilistic, statistical, and stochastic modeling procedures, which are explained with MATLAB programs in Chaps. 6, 7, and 10. The non-recording rain gauge provides total amount of rainfall as daily, weekly, monthly, or annual multi-rainfall events. Figure 2.1 represents a long-term annual rainfall record graphical representation in the form of time series from New Jersey, USA, during 1895–2013 periods.

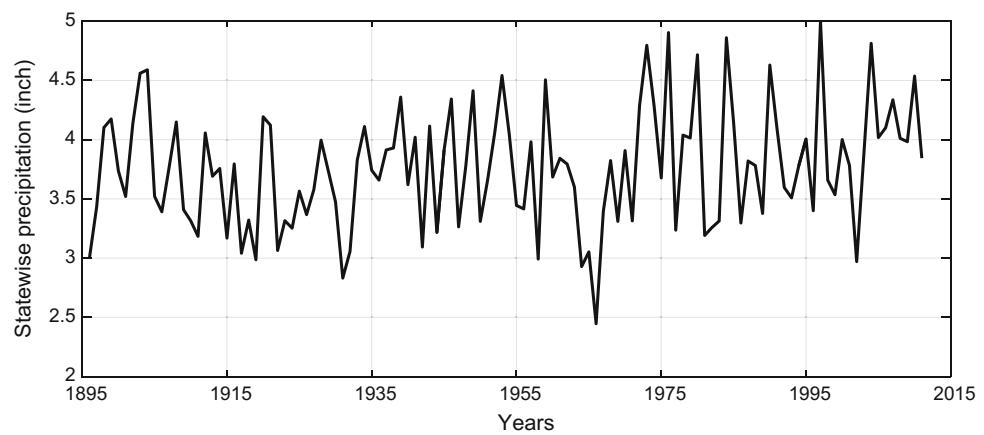
In general, annual records may include deterministic holistic monotonic or partial trends, shifts and uncertainty parts, which can be expressed by probabilistic, statistical, and stochastic approaches (Chap. 6).

In practical engineering applications, the most important feature of any storm rainfall event is its intensity, which is the speed of rainfall occurrence from beginning to the end of an event. The recording rain gauges provide the accumulative rainfall amount variation during each individual storm rainfall event in seconds or minutes as in Fig. 2.2, which is recorded on January 26, 2011, at Jeddah, Kingdom of Saudi Arabia (Sen 2015).

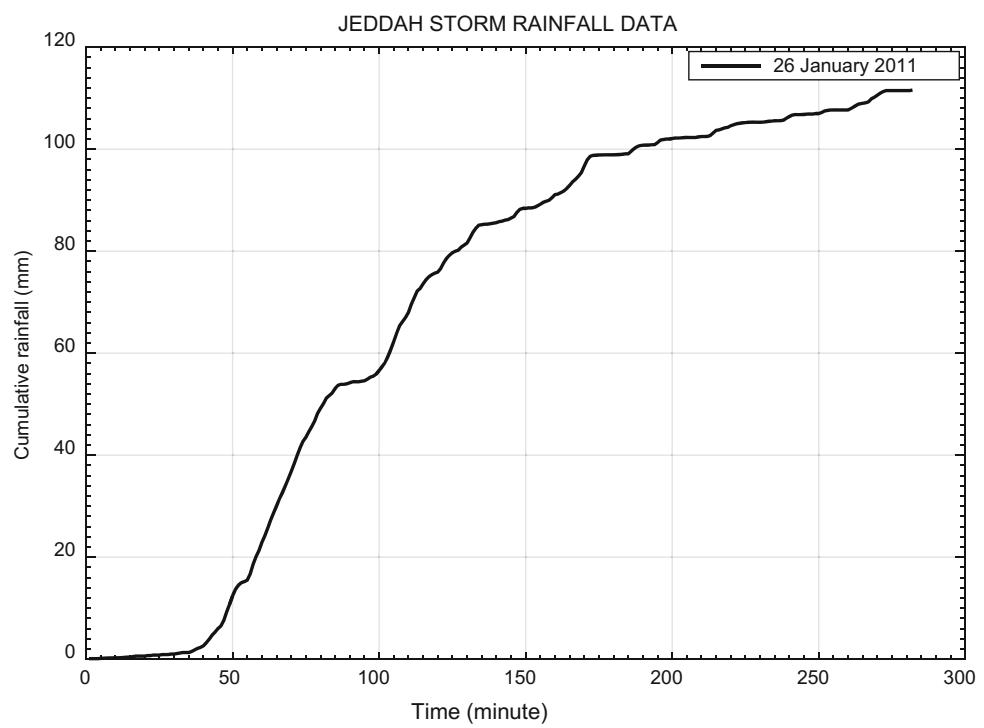
In practical applications, there are numerous benefits from the graphs in Figs. 2.1 and 2.2. The former is very useful to determine the extreme values (high and low peaks) for flood and drought calculations and their return periods. The latter paves the way to the fundamentals of rainfall intensity-duration-frequency (IDF) curves for various surface water design problems (Chap. 3) and groundwater recharge calculations (Chap. 4). Since each meteorological record has uncertainty, their future behaviors can be predicted based on the probabilistic, statistical, or stochastic methods under the concepts of uncertainty and risk.

The intensity-duration (ID) curve calculation is based on a given single storm record similar to Fig. 2.2 by means of the following software in MATLAB programming language.

**Fig. 2.1** New Jersey annual rainfall sequence



**Fig. 2.2** Single storm rainfall event charts



```

function IntensityDuration(X)
% This program calculates intensity-duration (ID) curve for a given
% single rainfall record
% X      : Two dimensional data (timexcumulative rainfall record)
% Time records in successive minutes (1,2,3,.....,.....)
% Cumulative rainfall amounts in mm
n=length(X(:,2)); % Data number
D=[5 10 20 50 70 100 120]; % Duration limits
for i=1:7
    d=D(i); % Duration
    nD=n-d+1; % Number of possible duration
    k=0;
    for j=1:nD
        k=k+1;
        ll=j;      % Lower limit of duration
        ul=j+d-1; % Upper limit of the duration
        I(k)=(X(ul,2)-X(ll,2))*60/d; % Possible intensities
    end
    Imax(i)=max(I); % Maximum intensity among all possible intensities
end
loglog(D,Imax,'r*')
grid on
box on
axis equal
end

```

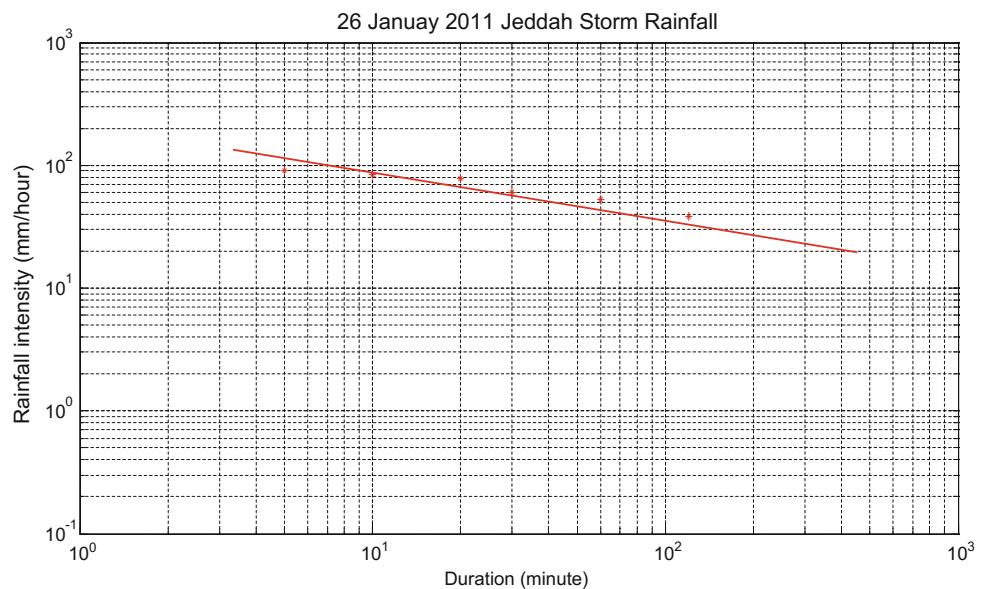
The application of this software to the cumulative rainfall data in Fig. 2.2 yields the ID curve on double-logarithmic paper as in Fig. 2.3.

### 2.2.1 Frequency (Risk) Calculation from Probability Distribution Function (PDF)

After the calculation of ID curve, probabilistic and statistical frequency methodologies help to construct theoretical

IDF curves. For the frequency calculations, a theoretical probability distribution function (PDF) must be matched to rainfall amounts of given duration (Sen 2008). Among the most frequently used PDFs in hydro-meteorological studies are generalized extreme value (GEV), Gumbel, normal (Gaussian), two-parameter log-normal, three-parameter log-normal, Gamma, Pearson Type III, and log-Pearson Type III PDFs. The mathematical expression,  $f(i)$ , for GEV PDF is given below for rainfall intensities at a particular time interval during the storm rainfall duration (Chow 1964).

**Fig. 2.3** Intensity–duration (ID) curves



$$f(i) = \frac{1}{\alpha} \left[ 1 - k \left( \frac{i-u}{\alpha} \right) \right]^{1/k-1} e^{-[1-k(\frac{i-u}{\alpha})]^{1/k}} \quad (2.1)$$

where  $\alpha$ ,  $u$ , and  $k$  are the model parameters. If the storm rainfall duration is, say, 180 min, then there are  $180/5 = 36$  non-overlapping 5-min intensities, which correspond to the storm rainfall record slope during this time interval. The intensity is defined as the rainfall increment amount divided by the specified time interval.

The extreme value Type I Gumbel PDF has the following mathematical expression again with parameters ( $\alpha$  and  $u$ ), which define the location and scale of the PDF shape (Gumbel 1958),

$$f(i) = \frac{1}{\alpha} \exp \left[ \left( -\frac{i-u}{\alpha} \right) - \exp \left( -\frac{i-u}{\alpha} \right) \right] \quad (2.2)$$

The earliest application of the normal (Gaussian) PDF to hydro-meteorological variables is presented by Hazen (1914). This PDF has symmetrical bell shape, and its general expression can be found in any standard textbook on statistics (Benjamin and Cornell 1970),

$$f(i) = \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-1}{2\sigma^2}(i-\mu)^2} \quad (2.3)$$

where  $\mu$  and  $\sigma$  are the arithmetic mean and standard deviation parameters of the PDF, respectively.

As one of the representative skewed PDFs, two-parameter log-normal PDF has the following mathematical formulations with logarithmic mean and standard deviation parameters,  $\mu_i$  and  $\sigma_i$ , respectively (Yevjevich 1972).

$$f(i) = \frac{1}{i \sigma_i \sqrt{2\pi}} \exp \left[ -\frac{(\ln i - \mu_i)^2}{2\sigma_i^2} \right] \quad (2.4)$$

Alternatively, the three-parameter log-normal distribution is similar except that  $i$  is shifted by an amount,  $m$ , which represents a lower bound of the PDF (Kite 1977),

$$f(i) = \frac{1}{(i-m)\sigma_i \sqrt{2\pi}} \exp \left\{ -\frac{[\ln(i-m) - \mu_i]^2}{2\sigma_i^2} \right\} \quad (2.5)$$

A flexible PDF that includes the exponential PDF (when  $\beta = 1$ ) is a two-parameter Gamma PDF that can be expressed mathematically as follows, where  $\alpha$  and  $\beta$  are parameters (Kite 1977).

$$f(i) = \frac{1}{\alpha^\beta \Gamma(\beta)} i^{\beta-1} e^{-(i/\alpha)} \quad (2.6)$$

Pearson (1930) suggested the Pearson III and then its logarithmic version, Log-Pearson III PDFs as,

$$f(i) = \frac{1}{\alpha \Gamma(\beta)} \left( \frac{i-\gamma}{\alpha} \right)^{\beta-1} e^{-\left( \frac{i-\gamma}{\alpha} \right)} \quad (2.7)$$

and

$$f(i) = \frac{1}{\alpha \Gamma(\beta)} \left[ \frac{\ln(i) - \gamma}{\alpha} \right]^{\beta-1} e^{-\left\{ \frac{\ln(i) - \gamma}{\alpha} \right\}} \quad (2.8)$$

respectively.

In the derivation of IDF curves, each PDF provides the following basic information for meaningful interpretations.

- (1) The form of the PDF shows the rainfall frequency as it is symmetrically distributed or skewed. For instance, in arid and semiarid regions, it is climatologically meaningful to expect storm rainfalls at high (low) intensities with low (high) frequencies. The theoretical PDF of such a statement abides by an exponential or Gamma PDF,
- (2) The PDFs give way to calculate the risk,  $R$ , values corresponding to a set of engineering design (return) periods,  $T$ , such as 2-year, 5-year, 10-year, 25-year, 50-year, 100-year, and 500-year.

Risk is the amount of uncertainty, because nature breaks records against engineering structural designs, which are expected to resist extreme hydro-meteorology events as runoff, flood, flash flood, and inundation occurrences. Whatever the refinement of the scientific methodology, there is always an unexplainable uncertainty part, and therefore, the designer has to take a certain risk level in his/her calculations. It is possible to predict design magnitude after acceptance of a certain risk level. Even after the inclusion of risk in the calculations with most modern methodologies, it is not guaranteed that in future engineering designs will not be toppled. Risk concept provides that extreme occurrences will not cause extreme destructions and dangers to property and human life. In the hydro-meteorological calculations, the most recommended risk level is, in general, 10% percent or for more sensitive works it may be adapted as 5%. It is useful to relate the risk (frequency),  $R$ , to the design (return) period,  $T$ , which are inversely related to each other as,

$$R = \frac{1}{T} \quad (2.9)$$

The frequency is also defined as the frequency of exceedence once during the whole design period.

The selection of suitable PDFs can be achieved by means of any available ready software. However, below simple PDF determination software is presented in MATLAB programming language.

```

function [VI,I,pr] = PDFselection(D,StationName)
% This software is written by Zekai Sen on 26 November 2016
% The main purpose is to match the most suitable Probability Distribution Function (PDF)
% to each variable.
% In this software Gamma, Log-Normal, Extreme Value (EV Gumbel) and Generalized
% Extreme Value (GEV, Pearson) PDF's are considered
% D : Data (nxND)
% I : Probability distribution function names
% If I = 1 Gamma PDF
% If I = 2 Log-Normal PDF
% If I = 3 Extreme value (Gumbel) PDF
% If I = 4 Generalized extreme value (Pearson III) PDF
% If I = 5 Weibull PDF
N=length(D);
Risk=0.001:0.001:0.999;
R=[1-Risk(500) 1-Risk(200) 1-Risk(100) 1-Risk(40) 1-Risk(20) 1-Risk(10) 1-Risk(4) 1-
Risk(2)];
Dmin=min(D); % Maximum data value
Dmax=max(D); % Minimum data value
x=Dmin:0.1:1.1*Dmax; % Data variation domain
pp=(1:1:N)/(N+1); % Data empirical probability (ascending order)
p=1-pp'; % Data probability in descending order
SV=sort(D); % Sorted data in ascending order
pgam=gamfit(D); % Gamma PDF parameters
ygam=1-gamcdf(x,pgam(1),pgam(2));
ptgam=1-gamcdf(SV,pgam(1),pgam(2));
ppt2gam=(p-ptgam).^2;
GTest=mean(ppt2gam);
plon=lognfit(D); % Log-Normal PDF parameters
ylon=1-logncdf(x,plon(1),plon(2));
ptlon=1-logncdf(SV,plon(1),plon(2));
ppt2lon=(p-ptlon).^2;
LNTTest=mean(ppt2lon);
pevd=evfit(D); % Gumbel PDF parameters
yevd=1-evcdf(x,pevd(1),pevd(2));
ptevd=1-evcdf(SV,pevd(1),pevd(2));
ppt2evd=(p-pteved).^2;
EVTest=mean(ppt2evd);
pgev=gevfit(D); % Pearson III PDF parameters
ygev=1-gevcdf(x,pgev(1),pgev(2),pgev(3));
ptgev=1-gevcdf(SV,pgev(1),pgev(2),pgev(3));
ppt2gev=(p-ptgev).^2;
GEVTest=mean(ppt2gev);
pwbl=wblfit(D); % Weibull PDF parameters
ywbl=1-wblcdf(x,pwbl(1),pwbl(2));
ptwbl=1-wblcdf(SV,pwbl(1),pwbl(2));
ppt2wbl=(p-ptwbl).^2;
WBLTest=mean(ppt2wbl);
rgam=gaminv(R,pgam(1),pgam(2));
rlon=logninv(R,plon(1),plon(2));
revd=evinv(R,pevd(1),pevd(2));
rgev=gevinv(R,pgev(1),pgev(2),pgev(3));
rwbl=wblinv(R,pwbl(1),pwbl(2));
[VI I]=min([GTest LNTTest EVTest GEVTest WBLTest]);
if I == 1
    yf=ygam;
    rf=rgam;
    pr=pgam;
    PR='Gamma PDF';
elseif I ==2
    yf=ylon;
    rf=rlon;
    pr=plon;
    PR='Log-normal PDF';
elseif I == 3
    yf=ylon;
    rf=rlon;
    pr=plon;
    PR='Extreme value (Gumbel) PDF';
elseif I == 4
    yf=ylon;
    rf=rlon;
    pr=plon;
    PR='Generalized extreme value (Pearson III) PDF';
elseif I == 5
    yf=ylon;
    rf=rlon;
    pr=plon;
    PR='Weibull PDF';
end

```

.....Continued.....

```

yf=yevd;
rf=revd;
pr=pevd;
PR='Gumbel PDF';
elseif I == 4
    yf=ygev;
    rf=rgev;
    pr=pgev;
    PR='Pearson PDF';
else
    yf=ywbl;
    rf=rwbl;
    pr=pwbl;
    PR='Weibull PDF';
end
figure
scatter(SV,p,'k*')
title(StationName)
hold on
grid on
box on
xlabel('Ground station rainfall values (mm)')
ylabel('Exceedence probability')
plot(x,yf,'LineWidth',2,'Color','r') % Theoretical PDF
legend('Data values',PR,'Location','Northeast')
text(Dmax/2,0.80,['Location constant = ' num2str(pr(1))])
text(Dmax/2,0.75,['Scale constant = ' num2str(pr(2))])
text(0.40*Dmax,0.60,[' 2-year (%50 risk) = ' num2str(rf(1)), ' mm'])
text(0.40*Dmax,0.55,[' 5-year (%20 risk) = ' num2str(rf(2)), ' mm'])
text(0.40*Dmax,0.50,[' 10-year (%10 risk) = ' num2str(rf(3)), ' mm'])
text(0.40*Dmax,0.45,[' 25-year (%4 risk) = ' num2str(rf(4)), ' mm'])
text(0.40*Dmax,0.40,[' 50-year (%2 risk) = ' num2str(rf(5)), ' mm'])
text(0.40*Dmax,0.35,['100-year (%1 risk) = ' num2str(rf(6)), ' mm'])
text(0.40*Dmax,0.30,['250-year (%0.4 risk) = ' num2str(rf(7)), ' mm'])
text(0.40*Dmax,0.25,['500-year (%0.2 risk) = ' num2str(rf(8)), ' mm'])
end

```

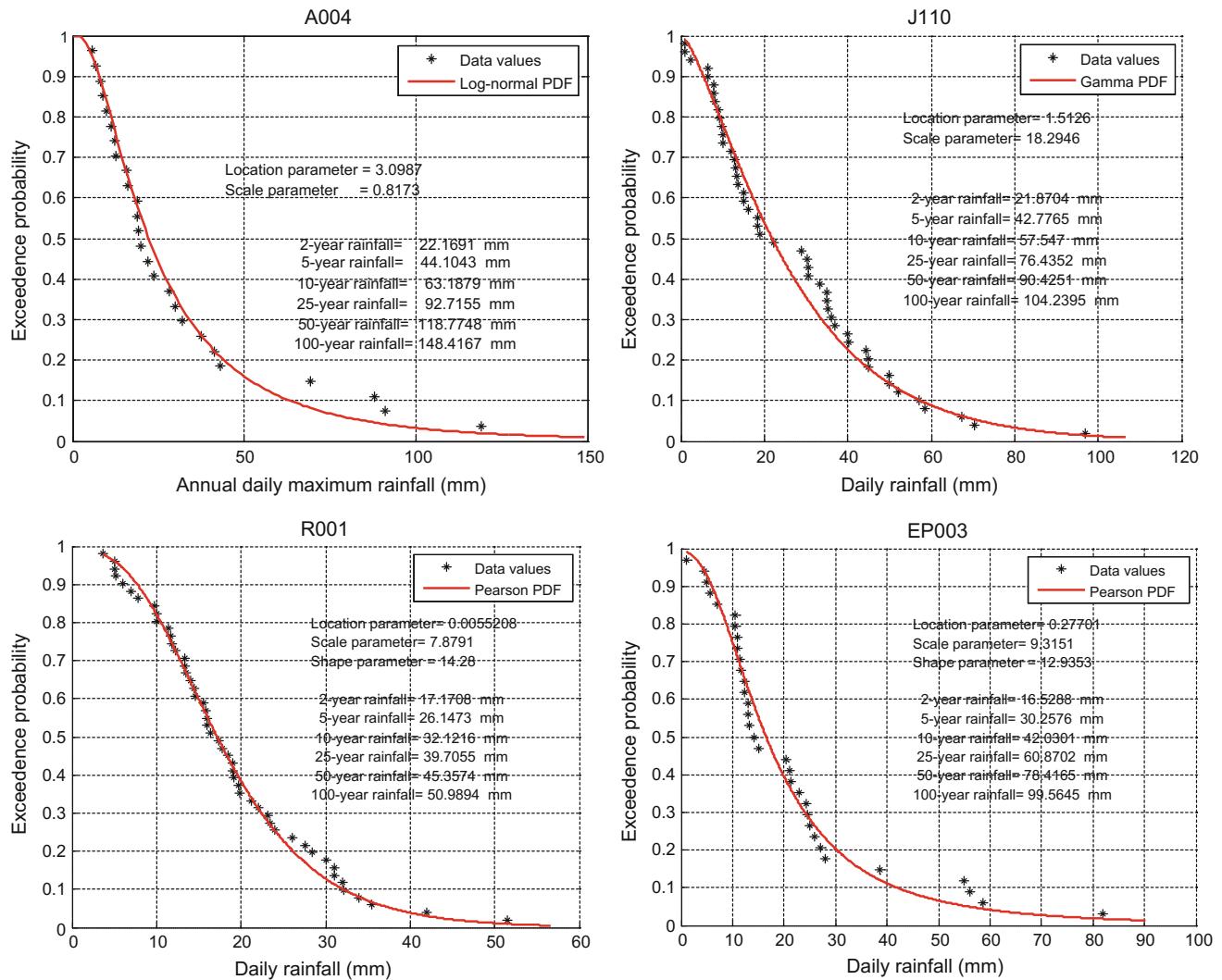
As an example, four meteorology station annual daily rainfall records are taken into consideration from different parts of the Arabian Peninsula, Kingdom of Saudi Arabia (Sen 2015). The full report is available at the Saudi Geological Survey (SGS). In the southwestern part at Asir Region, A004; Red Sea Coastal Region Makkah Al-Mukarramah City, J110; Riyadh Region, R101; and within the Eastern Region, EP003 stations are treated with the above software leading to results in Fig. 2.4.

In each graph, the most suitable type of the PDF to data scatter points is given with its parameters and return periods (2-year, 5-year, 10-year, 25-year, 50-year, and 100-year) corresponding to frequency (risk) levels (0.50, 0.20, 0.10, 0.04, 0.02, and 0.01).

### 2.3 Intensity–Duration–Frequency (IDF)

The knowledge of rainfall IDF triple relationship is of fundamental importance in hydro-meteorology, water resources systems design and management. IDF curves are based on records from recording gauges, which are also necessary prerequisites for many hydro-meteorological models and procedures for water quantity and quality computations (Chaps. 4 and 5).

IDF charts are essential ingredients to characterize storm rainfall events, and especially, engineering design period for any water structure dimensioning study. Storm rainfall events and consequent floods may constitute one of the most



**Fig. 2.4** Annual daily maximum rainfall cumulative PDFs for each station

striking illustrations of what is an extreme event. Risk evaluation and mitigation necessitate statistical information in order to plan appropriate infrastructures related to sewage, culvert, bridge, dam, dike, design, etc., in order to protect the

society and the goods against extreme hydro-meteorological events.

In developing IDF curves from a set of recording rain gauge storm rainfall charts (see Fig. 2.2), probability and

statistical frequency analysis are employed, whereby an extreme rainfall series of given duration is described by a suitable PDF as mentioned in Sect. 2.2.1. The following MATLAB program provides classical IDF curve derivations

provided that data are available from a set of recording meteorology stations. This software has similar parts to IntensityDuration (X) software that is presented in Sect. 2.2.

```

function IntensityDurationFrequency(X)
% This program calculates intensity-Duration-Frequency (IDF) curve for a
% set of given recording rainfall charts
% X      : N dimensional data (timexcumulative rainfall record)
%           The first column is for time records in successive minutes
%           (1,2,3,...,...,...)
%           Other columns are for cumulative rainfall amounts in mm for
%           each storm rainfall records at the same station
% N      : The number of storm rainfall charts
n=length(X(:,1)); % Data number
D=[5 10 25 50 100 250 500]; % Duration limits
%
% Calculation of intensity for each duration
%
for i=1:N
    for j=1:7
        d=D(j); % Duration
        nD=n-d+1; % Number of possible duration
        L=0; % Duration number counter
        for k=1:nD
            L=L+1;
            ll=k;      % Lower limit of duration
            ul=k+d-1; % Upper limit of the duration
            I(L)=(X(ul,2)-X(ll,2))*60/d; % Possible intensities
        end
        Imax(i)=max(I); % Maximum intensity among all possible intensities
    end
end
%
% Probability distribution function fit to each duration intensity
%
%
% Calculation of risk levels for each duration
%
%
%
% Graphical representation of IDF curves on double logarithmic scale
%
loglog(D,Imax,'r*')
grid on
box on
axis equal
end

```

Another version of the IDF calculation with resultant risk values are calculated by use of the following MATLAB software.

among these records is labeled as the annual daily maximum rainfall (ADMR) record rainfall duration as one day, i.e., 1440-min in humid regions of the world. In arid and semi-

```

function [Risk] = IntensityDurationFrequencyCurve(Veri,vs)
% This program is written by Zekai Sen on 12 October 2012
% Veri is the data that includes cumulative rainfall amounts with time
% The first row includes time of record and the other columns include
% cumulative storm rainfall amounts. Its dimension is mxn, where m is the
% vector of desired set of durations as in this program
% [10 20 30 60 120 180 360 720] and n is the number of storm
% vs data number
% t is the vector of intensity calculations [10 20 30 40 50 60 120, 180]
nd=8; % Number of duration data, which is 8 in this case
R=[10 25 50 100 200 500]; % Return periods; the inverse is the frequency
x=1:1:720; % Horizontal axis minute division
for k=2:vs
    yy=spline(Veri(1,:),Veri(k,:),x);
    for i=1:nd
        D=Veri(1,i);
        Ndur=round(720/D);
        for j=1:Ndur
            Intensity(j)=(yy(j*D)-yy((j-1)*D+1))/D;
        end
        MaxIntensity(k-1,i)=max(Intensity(1:Ndur));
    end
end
for i=1:nd
    MaxIns=MaxIntensity(:,i);
    par=gamfit(MaxIns(MaxIns>0)); % Considers only positive numbers
    Maxsort=sort(MaxIns(MaxIns>0));
    Max=max(MaxIns(MaxIns>0));
    Min=min(MaxIns(MaxIns>0));
    xx=0.001:0.01:1;
    yy=gamcdf(xx,par(1),par(2));
    for j=1:6
        Risk(i,j)=60*10*(gaminv(1-1/R(j),par(1),par(2))); % multiplied
        % by 60 in order to convert mm/min to mm/hour; multiplication
        % by 10 is for conversion of mm to cm
    end
end
scatter(Veri(1,:),Risk(:,1))
hold on
scatter(Veri(1,:),Risk(:,2))
scatter(Veri(1,:),Risk(:,3))
scatter(Veri(1,:),Risk(:,4))
scatter(Veri(1,:),Risk(:,5))
scatter(Veri(1,:),Risk(:,6))
end

```

### 2.3.1 Dimensionless Intensity–Duration (DID) Curve

IDF curves are not available in many parts of the world, because recording rainfall instruments do not exist. Few or several of them are available at a set of locations, and they provide general information about the region. In practice, most often a set of daily total rainfall amounts is available at any non-recording meteorology station and the maximum

arid regions one cannot observe 24-h long rainfall events, because they have shorter durations depending on the geographical location, which may be 3-h (180-min) or 6-h (360-min) rainfall representative durations. Whatever the duration, one can obtain the theoretical PDF of the ADMR data and then one is able to calculate a set of exceedence probability rainfall amounts for a given duration with a return period (or risk level). Hence, the question is how to relate ADMR records for IDF curve generation? In this

section, an innovative concept and methodology are presented through the following three steps for IDF curve establishment from the ADMR records.

- (1) The first step is to determine the most suitable PDF for the ADMR records,
- (2) The next step is to obtain dimensionless intensity duration (DID) curve for the region, where there may be some recording rain gauge graphs from which ID curve can be obtained as explained in Sect. 2.2. DID curve can be obtained by dividing durations of the ID curve by the maximum duration and intensities by the maximum intensity,
- (3) The third step is to convert the ADMR records to IDF curves through the DID curve, just by the reverse operation of division this time by multiplication of the DID dimensionless durations by the desired duration and the dimensionless intensities by the corresponding intensity value. This innovative methodology can be applied in any part of the world provided that DID curves are determined either from a set of already existing storm rainfall records or from a hypothetical DI curve.

The following recommendations must be taken into consideration for future application.

- (1) The IDF curves at each station must be updated after 5 or at the maximum 10 years with revision,
- (2) If storm rainfall records become available, then the IDF curves must be checked against the ones derived from the ADMR data. In general, not too many changes are expected, but the climate change impact may play some role (Chap. 7),
- (3) For local engineering structure design applications, the nearest meteorology station or the nearest 2–3 stations' IDF curves can be considered for regional DID representation purposes,
- (4) For drainage basin flood calculations, one should consider all the stations within the study area, if available, otherwise the nearest stations outside the drainage area can be considered by some techniques, such as the inverse distance square methodology or spatial dependence function method (Chap. 8),
- (5) In any engineering structure design, a relative error should be accepted and 10% is recommended.

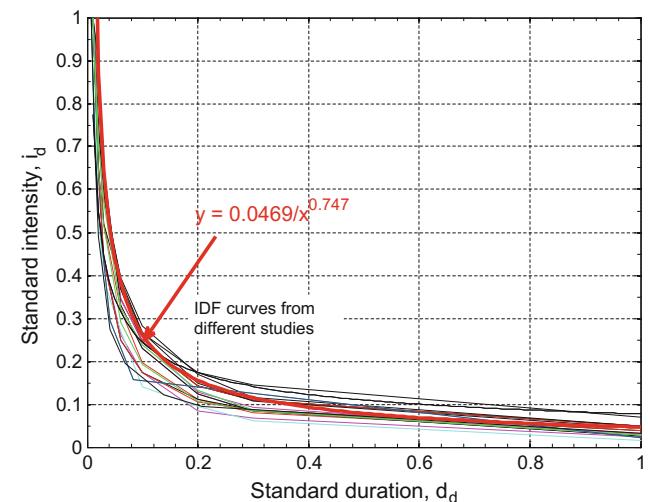
One can achieve IDF curves generation from the ADMR amounts and their combination with the DID through the execution of the following steps.

- 1 Since ADMR amounts are available for some daily duration (6-h = 360-min, 12-h = 720-min, 24-h = 1440 min), first the most suitable PDF is determined for each station,
- 2 The theoretical PDFs help to calculate rainfall amounts that correspond to a set of desired probabilities of exceedence (frequency, risk),  $R$ , according to the inverse relationship between the return period,  $T$ , and the risk in Eq. (2.9). The return period (risk) sets are adapted as  $T = 2\text{-year}$  ( $R = 0.50$ ),  $T = 5\text{-year}$  ( $R = 0.20$ ),  $T = 10\text{-year}$  ( $R = 0.10$ ),  $T = 25\text{-year}$  ( $R = 0.04$ ),  $T = 50\text{-year}$  ( $R = 0.02$ ), and  $T = 100\text{-year}$  ( $R = 0.01$ ),
- 3 The IDF curves can be generated on the basis of the available information from the previous steps concerning ADMR data and DID curve as explained above. Additionally, one should know rainfall amount corresponding to the exceedence probability for each return period from the suitable PDF for each station,
- 4 Available natural IDF curves from different parts of a region or country can be converted to DID curves with the best average DID curve expression as in Fig. 2.5,

$$i_d = \frac{0.0469}{d_d^{0.747}} \quad (2.10)$$

where  $i_d$  and  $d_d$  are dimensionless intensity and duration, respectively,

- 5 Since the definition of DID curve is the change of dimensionless intensity by dimensionless time,  $d_d$ , it is



**Fig. 2.5** Dimensionless intensity–duration curves

possible to calculate rainfall duration,  $t_r$ , for the IDF graph through the following expressions,

$$t = t_r d_d \quad (2.11)$$

On the other hand, considering DID curve expression from Eq. (2.10) with the rainfall amounts,  $r_e$ , the intensity,  $i$ , is calculated as,

$$i = i_i r_e \quad (2.12)$$

The software to obtain the DID curve from a single or a set of IDF curves is presented in the following MATLAB program.

### 2.3.2 PDF and IDF Curve Software

The first step in any IDF curve calculation as mentioned in Sect. 2.2 is the theoretical PDF determination that fits the available ADMR data best. The software below fits the best PDF among the most commonly used PDFs in hydro-meteorology domain. In the software, 5 PDFs are Gamma, log-normal, extreme value (Gumbel), generalized extreme value (Pearson III), and Weibull theoretical expressions. The software statements are all in MATLAB language, which are rather easy to understand with subsequent comment statements for explanations. After the suitability of the theoretical PDF for each station, one can then generate the IDF curves for desired daily rainfall duration, say, 360-min by the following software.

```

function DimensionlessIntensityDurationCurve(X)
% This program calculates Dimensionless-Intensity-Duration(DID) curve for a
% set of given recording rainfall charts
% X      : N dimensional data (timexcumulative rainfall record)
%           The first column is for time records in successive minutes
%           (1,2,3,.....)
%           Other columns are for cumulative rainfall amounts in mm for
%           each storm rainfall records at the same station
% N      : The number of storm rainfall charts
n=length(X(:,1)); % Data number
D=[5 10 25 50 100 250 500]; % Duration limits
%
% Calculation of intensity for each duration
%
for i=1:N
    for j=1:7
        d=D(j); % Duration
        nD=n-d+1; % Number of possible duration
        L=0; % Duration number counter
        for k=1:nD
            L=L+1;
            ll=k; % Lower limit of duration
            ul=k+d-1; % Upper limit of the duration
            I(L)=(X(ul,2)-X(ll,2))*60/d; % Possible intensities
        end
        Imax(i)=max(I); % Maximum intensity among all possible intensities
    end
%
% Probability distribution function fit to each duration intensity
%
%
% Calculation of risk levels for each duration
%
%
% Graphical representation of IDF curves on double logarithmic scale
%
loglog(D,Imax,'r*')
grid on
box on
axis equal
end

```

```

function [DI] = IDFcurveDailyMaxRainSixPDFsAndTable(D,DesiredDuration,StName)
% BE CAREFUL FOR WHICH REGION OR COUNTRY THE EQUATION ON LINE 40 AND % 41??
% This program is written on 13 September 2015 Sunday by Zekai Sen
% Gamma, Log-Normal, Extreme Value and Generalized
% Extreme Value probability distribution functions are considered
% D : Daily maximum rainfall data for each year
% DesiredDuration : Total duration of the IDF curves
% StName : Station name
% V : It is the least sum of squares of probability deviations
% from the theoretical probability distribution
% I : The number of PDF
% If I = 1 Gamma PDF
% If I = 2 Log-Normal PDF
% If I = 3 Extreme value (Gumbel) PDF
% If I = 4 Generalized extreme value (Pearson III) PDF
% If I = 5 Weibull PDF
% If I = 6 Logarithmic Generalized extreme value (Log-Pearson III) PDF
n=length(D);
DM=1.1*max(D);
Dm=min(D);
x=Dm:0.1:DM;
pp=(1:1:n)/(n+1); % Data probability in ascending order
p=1-pp'; % Data probability in descending order
SD=sort(D); % Sorted time series in ascending order
pgam=gamfit(D); % Gamma PDF parameters
ygam=1-gamcdf(x,pgam(1),pgam(2));
ptgam=1-gamcdf(SD,pgam(1),pgam(2));
ppt2gam=(p-ptgam).^2;
GTest=mean(ppt2gam);
plon=lognfit(D); % Log-Normal PDF parameters
ylon=1-logncdf(x,plon(1),plon(2));
ptlon=1-logncdf(SD,plon(1),plon(2));
ppt2lon=(p-ptlon).^2;
LNTTest=mean(ppt2lon);
pevd=evfit(D); % Extreme value PDF parameters
yevd=1-evcdf(x,pevd(1),pevd(2));
pteved=1-evcdf(SD,pevd(1),pevd(2));
ppt2evd=(p-pteved).^2;
EVTest=mean(ppt2evd);
pgev=gevfit(D); % Generalized extreme value PDF parameters
ygev=1-gevcdf(x,pgev(1),pgev(2),pgev(3));
ptgev=1-gevcdf(SD,pgev(1),pgev(2),pgev(3));
ppt2gev=(p-ptgev).^2;
GEVTest=mean(ppt2gev);
pwbl=wblfit(D); % Weibull PDF parameters
ywbl=1-wblcdf(x,pwbl(1),pwbl(2));
ptwbl=1-wblcdf(SD,pwbl(1),pwbl(2));
ppt2wbl=(p-ptwbl).^2;
WBLTest=mean(ppt2wbl);
LogD=log(D); % Logarithmic data values
plgev=gevfit(LogD); % Generalized extreme value PDF parameters
ylgev=1-gevcdf(x,plgev(1),plgev(2),plgev(3));
ptlgev=1-gevcdf(SD,plgev(1),plgev(2),plgev(3));
ppt2lgev=(p-ptlgev).^2;
LGEVTest=mean(ppt2lgev);
% Maximum daily rainfall calculation for given return periods for each PDF
R=[2 5 10 25 50 100];
rgam=gaminv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],pgam(1),pgam(2));
rlon=logninv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],plon(1),plon(2));
revd=evinv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],pevd(1),pevd(2));
rwbl=wblinv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],pwbl(1),pwbl(2));
rgev=gevinv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],pgev(1),pgev(2),pgev(3));
rlgev=gevinv([1-1/R(1) 1-1/R(2) 1-1/R(3) 1-1/R(4) 1-1/R(5) 1-1/R(6)],plgev(1),plgev(2),plgev(3));
[V I]=min([GTest LNTTest EVTest GEVTest WBLTest LGEVTest]);

```

.....Continued.....

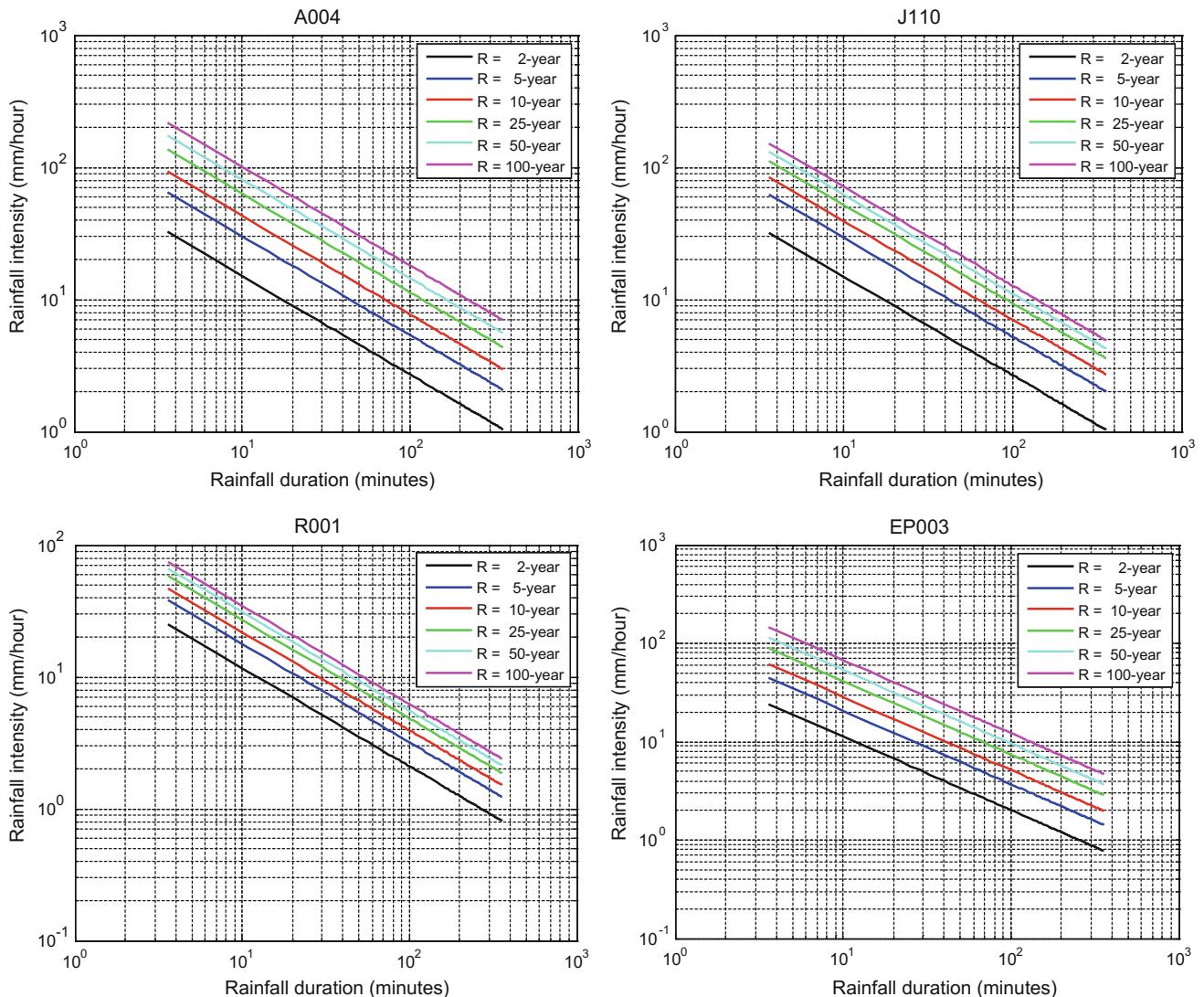
```

if I == 1
    yf=ygam;
    rf=rgam;
    pr=pgam;
    PDF='Gamma PDF';
elseif I ==2
    yf=ylon;
    rf=rlon;
    pr=plon;
    PDF='Log-normal PDF';
elseif I == 3
    yf=yevd;
    rf=revd;
    pr=pevd;
    PDF='Extreme value (Gumbel) PDF';
elseif I == 4
    yf=ygev;
    rf=rgev;
    pr=pgev;
    PDF='Generalized extreme value (Pearson) PDF';
elseif I == 5
    yf=ywbl;
    rf=rwbl;
    pr=pwbl;
    PDF='Weibull PDF';
else
    yf=ylgev;
    rf=rlgev;
    pr=plgev;
    PDF='Logarithmic Generalized extreme value (Log-Pearson) PDF';
end
figure
scatter(SD,p,'k*')
hold on
grid on
xlabel('Daily rainfall (mm)')
ylabel('Exceedence probability')
plot(x,yf,'LineWidth',2,'Color','r') % Theoretical PDF plot
legend('Data values',PDF,'Location','Northeast')
text(DM/2,0.80,['Location parameter= ' num2str(pr(1))])
text(DM/2,0.75,['Scale parameter= ' num2str(pr(2))])
if I == 4
    text(DM/2,0.70,['Shape parameter = ' num2str(pr(3))])
else
end
title(StName)
text(DM/2,0.60,[' 2-year rainfall= ' num2str(rf(1)), ' mm'])
text(DM/2,0.55,[' 5-year rainfall= ' num2str(rf(2)), ' mm'])
text(DM/2,0.50,[' 10-year rainfall= ' num2str(rf(3)), ' mm'])
text(DM/2,0.45,[' 25-year rainfall= ' num2str(rf(4)), ' mm'])
text(DM/2,0.40,[' 50-year rainfall= ' num2str(rf(5)), ' mm'])
text(DM/2,0.35,['100-year rainfall= ' num2str(rf(6)), ' mm'])
st=0:0.01:1; % standard time
RainDuration=DesiredDuration*st; %Required duration is 6 hours (360 minute)
% DISCRETE DURATION CALCULATION
DD(1)=ceil(10*101/360);
DD(2)=ceil(20*101/360);
DD(3)=ceil(30*101/360);
DD(4)=ceil(60*101/360);
DD(5)=ceil(120*101/360);
DD(6)=ceil(180*101/360);
DD(7)=ceil(360*101/360);
%sr=0.04./st.^0.5; % DID formulation valid for the Asian and European sides of Istanbul,
Turkey
sr=0.0308./st.^0.756; % DID formulation valid for the South-eastern part of Turkey
(Ceylanpinar)

```

.....Continued.....

```
% sr=0.0469./st.^0.747; % DID formulation valid for the Kingdom of Saudi Arabia
figure
Intensity=rf(1)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','k','MarkerSize',2)
% DISCRETE DURATION CALCULATION
DI2(1)=Intensity(DD(1));
DI2(2)=Intensity(DD(2));
DI2(3)=Intensity(DD(3));
DI2(4)=Intensity(DD(4));
DI2(5)=Intensity(DD(5));
DI2(6)=Intensity(DD(6));
DI2(7)=Intensity(DD(7));
hold on
Intensity=rf(2)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','b','MarkerSize',2)
% DISCRETE DURATION CALCULATION
DI5(1)=Intensity(DD(1));
DI5(2)=Intensity(DD(2));
DI5(3)=Intensity(DD(3));
DI5(4)=Intensity(DD(4));
DI5(5)=Intensity(DD(5));
DI5(6)=Intensity(DD(6));
DI5(7)=Intensity(DD(7));
Intensity=rf(3)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','r','MarkerSize',2)
% DISCRETE DURATION CALCULATION
DI10(1)=Intensity(DD(1));
DI10(2)=Intensity(DD(2));
DI10(3)=Intensity(DD(3));
DI10(4)=Intensity(DD(4));
DI10(5)=Intensity(DD(5));
DI10(6)=Intensity(DD(6));
DI10(7)=Intensity(DD(7));
Intensity=rf(4)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','g','MarkerSize',2)
% DISCRETE DURATION CALCULATION
DI25(1)=Intensity(DD(1));
DI25(2)=Intensity(DD(2));
DI25(3)=Intensity(DD(3));
DI25(4)=Intensity(DD(4));
DI25(5)=Intensity(DD(5));
DI25(6)=Intensity(DD(6));
DI25(7)=Intensity(DD(7));
Intensity=rf(5)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','c','MarkerSize',2)
% DISCRETE DURATION CALCULATION
DI50(1)=Intensity(DD(1));
DI50(2)=Intensity(DD(2));
DI50(3)=Intensity(DD(3));
DI50(4)=Intensity(DD(4));
DI50(5)=Intensity(DD(5));
DI50(6)=Intensity(DD(6));
DI50(7)=Intensity(DD(7));
Intensity=rf(6)*sr;
loglog(RainDuration,Intensity,'LineWidth',2,'Color','m','MarkerSize',2)
DI100(1)=Intensity(DD(1));
DI100(2)=Intensity(DD(2));
DI100(3)=Intensity(DD(3));
DI100(4)=Intensity(DD(4));
DI100(5)=Intensity(DD(5));
DI100(6)=Intensity(DD(6));
DI100(7)=Intensity(DD(7));
grid on
legend('R = 2-year','R = 5-year','R = 10-year','R = 25-year','R = 50-year','R = 100-year')
title(StName)
xlabel('Rainfall duration (minutes)')
ylabel('Rainfall intensity (mm/hour)')
DI=[DI2;DI5;DI10;DI25;DI50;DI100]; % Duration - Intensity table
end
```



**Fig. 2.6** Annual daily maximum rainfall IDF curves for each station

The IDF curves for the same set of stations (see Fig. 2.4) are presented in Fig. 2.6 on double-logarithmic axes.

The same software also yields the IDF curve tables (Table 2.1), which include demandable information concerning IDF curves for any water resources engineering design studies.

## 2.4 Radar Data

Radar reflections are used for rainfall estimation through a power relationship between the reflectivity,  $Z$ , and the rainfall intensity,  $R$ , as,

**Table 2.1** Annual daily maximum rainfall IDF tables for each station

A004		Duration (min)						
		10	20	30	60	120	180	360
Return period (year)	2	15.1175	9.0076	6.6538	3.9646	2.3623	1.745	1.0397
	5	30.0747	17.9197	13.2371	7.8872	4.6995	3.4715	2.0684
	10	43.0871	25.6731	18.9643	11.2997	6.7328	4.9735	2.9634
	25	63.2207	37.6695	27.8259	16.5798	9.879	7.2974	4.3481
	50	80.9891	48.2567	35.6465	21.2397	12.6555	9.3484	5.5702
	100	101.2002	60.2993	44.5422	26.5401	15.8137	11.6813	6.9602
J110		Duration (min)						
		10	20	30	60	120	180	360
Return period (year)	2	14.9137	8.8862	6.5641	3.9112	2.3304	1.7215	1.0257
	5	29.1699	17.3807	12.8388	7.6499	4.5581	3.367	2.0062
	10	39.2422	23.3821	17.2721	10.2914	6.132	4.5297	2.699
	25	52.1224	31.0567	22.9411	13.6693	8.1447	6.0164	3.5848
	50	61.6623	36.741	27.14	16.1712	9.6354	7.1176	4.2409
	100	71.0826	42.3539	31.2863	18.6416	11.1075	8.2049	4.8888
R001		Duration (mm)						
		10	20	30	60	120	180	360
Return period (year)	2	11.709	6.9767	5.1536	3.0707	1.8297	1.3515	0.8053
	5	17.8303	10.624	7.8478	4.676	2.7862	2.0581	1.2263
	10	21.9042	13.0514	9.6409	5.7445	3.4228	2.5284	1.5065
	25	27.0758	16.1329	11.9171	7.1007	4.2309	3.1253	1.8622
	50	30.93	18.4293	13.6135	8.1115	4.8332	3.5702	2.1273
	100	34.7705	20.7177	15.3039	9.1187	5.4333	4.0135	2.3914
EP003		Duration (min)						
		10	20	30	60	120	180	360
Return period (year)	2	11.2712	6.7159	4.9609	2.9559	1.7613	1.301	0.7752
	5	20.6332	12.2941	9.0815	5.4111	3.2242	2.3816	1.4191
	10	28.661	17.0774	12.6148	7.5164	4.4786	3.3083	1.9712
	25	41.5083	24.7324	18.2695	10.8857	6.4861	4.7912	2.8548
	50	53.4735	31.8617	23.5358	14.0236	8.3558	6.1723	3.6777
	100	67.8946	40.4544	29.8831	17.8056	10.6093	7.8369	4.6696

$$Z = aR^b \quad (2.13)$$

The parameters  $a$  and  $b$  depend on the precipitation type (i.e., drop size distribution) and temperature. In general, the reflectivity in dBZ (decibel) is converted to rainfall intensity (mm/h) as in any IDF curve graphs. The radar measurements

are recorded in a 3D polar coordinate system, where pixel position distances are available within an angle from the center. In a radar scan, each pixel has a reflectivity value depending on the range (distance), azimuth (angle), and elevation (height) axes. The raw reflectivity is converted to Cartesian coordinates at a specified height above ground

level for the next step. The reflectivity is then converted into rainfall intensity in mm/h by Eq. (2.13) with convenient constants. The rain rates are converted to 1-min accumulation data. The estimated radar rainfall is then adjusted against the values found from rain gauges surrounding the radar.

Apart from the above classical approach, in this section, radar reflection data conversion to ground rainfall amounts is achieved by converting the radar reflections' PDF to the ground meteorology station measurements PDF.

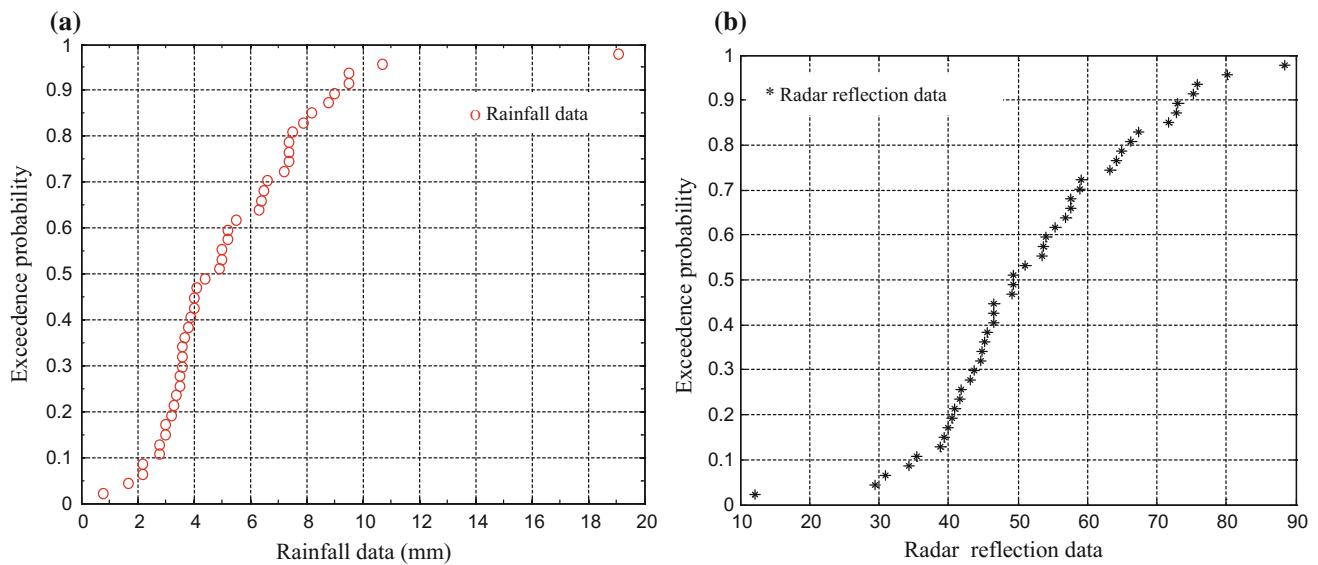
#### 2.4.1 PDFs Matching Procedure

As mentioned earlier in this chapter, any meteorological measurement includes uncertainties that can be dealt with the probability theory. The radar reflectivity measurements include the rainfall amounts in some way, which can be matched simultaneously with ground measurements. If the radar reflectivity and rainfall value PDFs are denoted

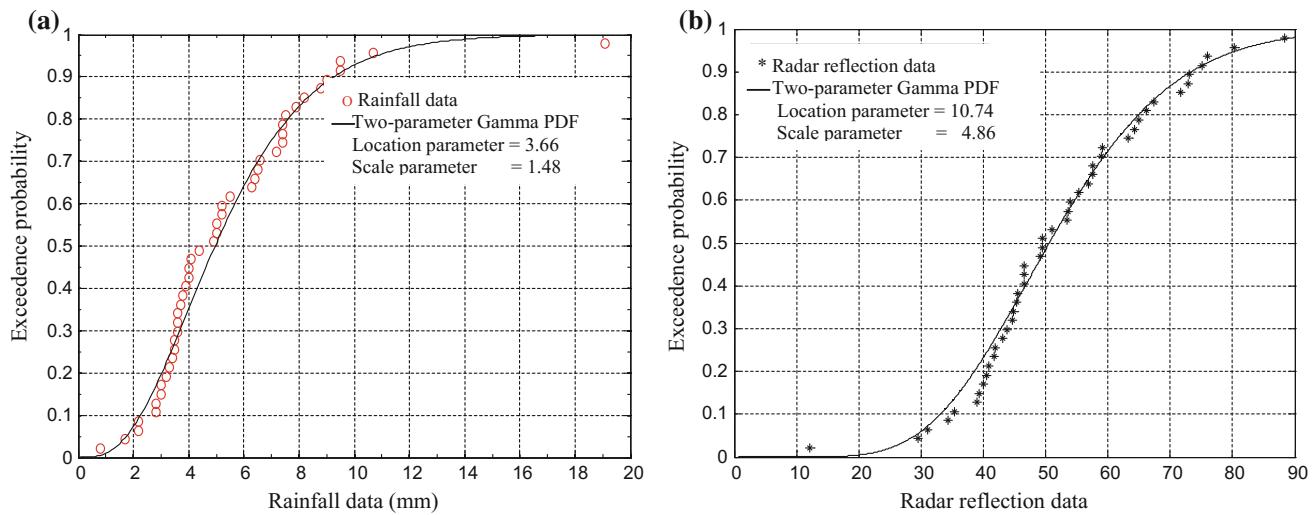
by  $f(Z)$  and  $f(r)$ , respectively, then for their matching procedure the following steps are necessary.

- (1) First obtain the theoretical PDFs or preferably CDFs of radar reflectivity and corresponding ground rainfall measurements as exemplified in Fig. 2.7,
- (2) Fit the most suitable theoretical CDFs to each one of the scatter diagrams (see Fig. 2.7). For this purpose, as mentioned in Sect. 2.2 try the most frequently used normal (Gaussian), log-normal, Gamma, Weibull, Gumbel, and Pearson III CDFs. For instance, in Fig. 2.8 the two-parameter theoretical CDFs appear,
- (3) Finally, these two CDFs are matched onto each other by shifting and scaling procedures with correspondence on the horizontal axis values only (see Fig. 2.9).

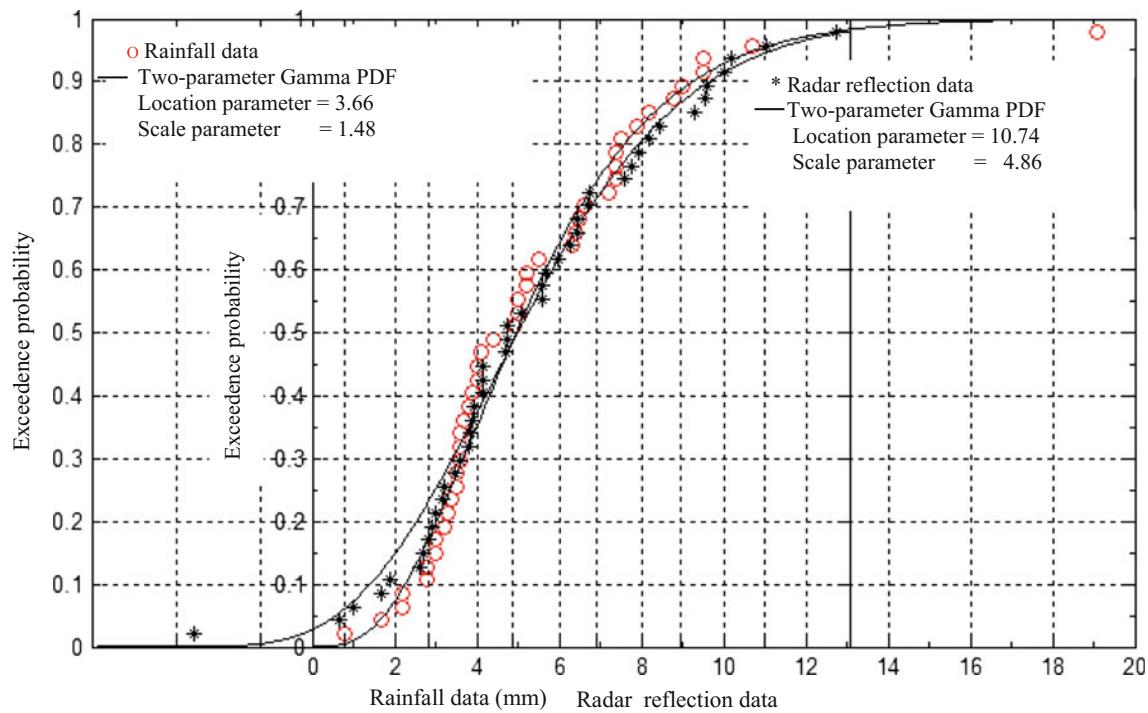
All the aforementioned explanations and graphs can be obtained as the outputs of the following two MATLAB software programs for the CDF search and for the PDFs match between the ground rainfall and radar reflectivity values, respectively.



**Fig. 2.7** Data CDF scatter points



**Fig. 2.8** The most suitable theoretical CDFs



**Fig. 2.9** CDFs match

```

function [I,SX,yy,pram,CDFTitle] = CDFit(X,XTitle)
% This program finds the best fit to a given series of data and it is
% written by Zekai Sen on 31 October, 2015 Saturday at 17:40 PM
% X       : Data series
% I       : Convenient CDF
% XTitle  : Horizontal axis title
% Various CDF considered in the program
%           I = 1, Normal CDF
%           I = 2, Logarithmic-normal CDF
%           I = 3, Exponential CDF
%           I = 4, Gamma CDF
%           I = 5, Weibull CDF
%           I = 6, Gumbel (Extreme value) CDF
%           I = 7, Pearson Type III (Generalized extreme value) CDF
% T=['Normal CDF','Log-Normal CDF','Exponential CDF','Gamma CDF','Weibull CDF','Gumbel CDF','Pearson CDF'];
n=length(X); % Number of data
p=(1:1:n)/(n+1);
SX=sort(X); % Sort X data
figure
% scatter(SX,1-p,'k*')
scatter(SX,p,'k*')
hold on
xlabel(XTitle)
ylabel('Exceedence probability')
grid on
box on
Xmin=min(X);
Xmax=max(X);
x=Xmin:0.01:1.1*Xmax;
% Theoretical normal CDF
prnorm(1)=mean(X);
prnorm(2)=std(X);
% y(1,1:n)=1-normcdf(SX,prnorm(1),prnorm(2));
y(1,1:n)=normcdf(SX,prnorm(1),prnorm(2));
Enorm=sum((p-y(1,1:n)).^2);
% Theoretical logarithmic-normal CDF
prlogn=lognfit(X);
% y(2,1:n)=1-logncdf(SX,prlogn(1),prlogn(2));
y(2,1:n)=logncdf(SX,prlogn(1),prlogn(2));
Elogn=sum((p-y(2,1:n)).^2);
% Theoretical exponential CDF
prexp=expfit(X);
% y(3,1:n)=1-expcdf(SX,prexp);
y(3,1:n)=expcdf(SX,prexp);
Eexp=sum((p-y(3,1:n)).^2);
% Theoretical Gamma CDF
prgam=gamfit(X);
% y(4,1:n)=1-gamcdf(SX,prgam(1),prgam(2));
y(4,1:n)=gamcdf(SX,prgam(1),prgam(2));
Egam=sum((p-y(4,1:n)).^2);
% Theoretical Weibull CDF
prwbl=wblfit(X);
% y(5,1:n)=1-wblcdf(SX,prwbl(1),prwbl(2));
y(5,1:n)=wblcdf(SX,prwbl(1),prwbl(2));
Ewbl=sum((p-y(5,1:n)).^2);
% Theoretical extreme value CDF
prev=evfit(X);
% y(6,1:n)=1-evcdf(SX,prev(1),prev(2));
y(6,1:n)=evcdf(SX,prev(1),prev(2));
Eev=sum((p-y(6,1:n)).^2);
% Theoretical generalized extreme value CDF
% prgev=gevfit(X);
% y(7,1:n)=1-gevcdf(SX,prgev(1),prgev(2),prgev(3));
% y(7,1:n)=gevcdf(SX,prgev(1),prgev(2),prgev(3));
% Egev=sum((p-y(7,1:n)).^2);
% plot(SX,y,'r')
%Find the best suitable PDF
.....Continued.....

```

```
% [V I]=min([Enorm Elogn Eexp Egam Ewbl Eev Egev]); % BE CAREFUL, IS IT min OR max?
[V I]=min([Enorm Elogn Eexp Egam Ewbl Eev]); % BE CAREFUL, IS IT min OR max?
yy(1,1:n)=y(I,1:n)';
plot(SX,yy,'r')
if I==1
    legend('Data values','Normal CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(prnorm(1))])
    text(Xmin,0.20,[' Shape parameter = ' num2str(prnorm(2))])
    pram=prnorm;
    CDFTitle='Normal CDF';
elseif I==2
    legend('Data values','Logarithmic normal CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(prlogn(1))])
    text(Xmin,0.20,[' Shape parameter = ' num2str(prlogn(2))])
    pram=prlogn;
    CDFTitle='Logarithmic normal CDF';
elseif I==3
    legend('Data values','Exponential CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(pexp(1))])
    pram=pexp;
    CDFTitle='Exponential CDF';
elseif I==4
    legend('Data values','Gamma CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(prgam(1))])
    text(Xmin,0.20,[' Shape parameter = ' num2str(prgam(2))])
    pram=prgam;
    CDFTitle='Gamma CDF';
elseif I==5
    legend('Data values','Weibull CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(prwbl(1))])
    text(Xmin,0.20,[' Shape parameter = ' num2str(prwbl(2))])
    pram=prwbl;
    CDFTitle='Weibull CDF';
elseif I==6
    legend('Data values','Gumbel - Extreme value CDF')
    text(Xmin,0.30,[' Scale parameter = ' num2str(prev(1))])
    text(Xmin,0.20,[' Shape parameter = ' num2str(prev(2))])
    pram=prev;
    CDFTitle='Gumbel - Extreme value CDF';
% elseif I==7
%     legend('Data values','Pearson III - Generalized extreme value CDF')
%     text(Xmin,0.30,[' Scale parameter = ' num2str(prgev(1))])
%     text(Xmin,0.20,[' Shape parameter1 = ' num2str(prgev(2))])
%     text(Xmin,0.10,[' Shape parameter2 = ' num2str(prgev(3))])
%     pram=prgev;
%     CDFTitle='Pearson III - Generalized extreme value CDF';
else
end
end
```

```

function [IY,IR,RYT] = RainfallMeasurementRadarReflectionPDFMatch(R,Y,StationName)
% This program is written by Zekai Sen on 26 November 2016
% Its purpose is to match radar reflection probability distribution
% function (PDF) to ground meteorology station PDF and then predict radar rainfall amounts
% R      = Radar reflection data
% Y      = Ground meteorology station data
% RYT    = Radar rainfall prediction values
R=R+0.001;
Y=Y+0.001;
N=length(R);
[VIR,IY,prY] = PDFselection(Y,'Ground station rainfall data (mm)');
[VIR,IR,prR] = PDFselection(R,'Radar reflection data (db)');
if IY == 1
    PY=1-gamcdf(Y,prY(1),prY(2));
    PDY='Gamma PDF';
elseif IY == 2
    PY=logncdf(Y,prY(1),prY(2));
    PDY='Log-normal PDF';
elseif IY == 3
    PY=evcdf(Y,prY(1),prY(2));
    PDY='Gumbel PDF';
elseif IY == 4
    PY=gevcdf(Y,prY(1),prY(2),prY(3));
    PDY='Pearson PDF';
else
    PY=wblcdf(Y,prY(1),prY(2));
    PDY='Weibull PDF';
end
if IR == 1
    PR=gamcdf(R,prR(1),prR(2));
    PDR='Gamma PDF';
elseif IR == 2
    PR=logncdf(R,prR(1),prR(2));
    PDR='Log-normal PDF';
elseif IR == 3
    PR=evcdf(R,prR(1),prR(2));
    PDR='Gumbel PDF';
elseif IR == 4
    PR=gevcdf(R,prR(1),prR(2),prR(3));
    PDR='Pearson PDF';
else
    PR=wblcdf(R,prR(1),prR(2));
    RYT=wblinv(PR,prY(1),prY(2));
    PDR='Weibull PDF';
end
if IY == 1
    RYT=gaminv(PR,prY(1),prY(2));
elseif IY == 2
    RYT=logninv(PR,prY(1),prY(2));
elseif IY == 3
    RYT=evinv(PR,prY(1),prY(2));
elseif IY == 4
    RYT=gevinv(PR,prY(1),prY(2),prY(3));
else
    RYT=wblinv(PR,prY(1),prY(2));
end
legend('Radar reflection data (db)',PDR,'Location','Southeast')
[VIRYT,IRYT,prRYT] = PDFselection(RYT,'Radar reflection prediction rainfall(mm)');
figure
scatter(sort(Y),sort(RYT),'k*')
xlabel('Ground station rainfall data (mm)')
ylabel('Radar reflection rainfall data(mm)')
title(StationName)
MY=max(Y);
MRYT=max(RYT);
M=max(MY,MRYT);
line([0 M], [0 M],'LineWidth',2)
grid on
box on
end

```

## 2.4.2 Application

The principles in the previous section are applied to Florya Meteorology Station, Istanbul, Turkey, rainfall simultaneous records with radar records as presented in Table 2.2.

The application of the steps in the previous section to the fourth and eight columns' radar reflectivity and ground rainfall values, respectively, leads to graphs in Fig. 2.10. The rainfall predictions according to the most suitable Gamma CDF are given in the last column of this table. The

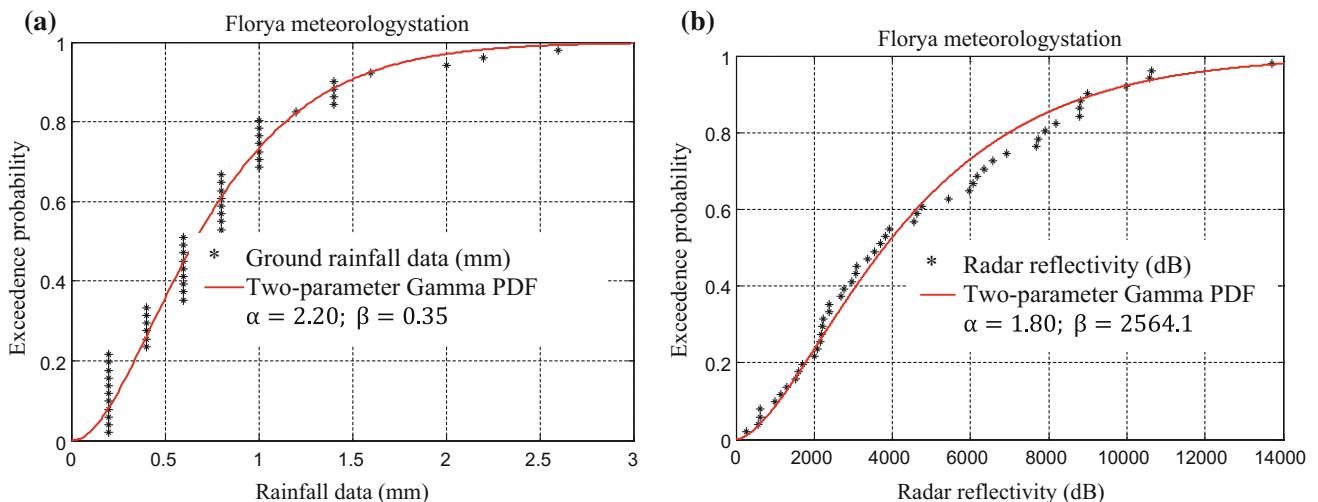
**Table 2.2** Radar reflectivity and meteorology station measurements

Time (h)	Row	Column	Reflectivity	Calculated data (mm)	Longitude	Latitude	Ground data (mm)	Calculated PDF data (mm)
1	2	3	4	5	6	7	8	9
14.04.2012 15:00	468	481	2773	0.2773	40.9748	28.7855	0.2	0.512
14.04.2012 15:06	468	481	7680	0.768	40.9748	28.7855	0.2	1.277
14.04.2012 15:12	468	481	1005	0.1005	40.9748	28.7855	0.6	0.212
14.04.2012 15:18	468	481	1303	0.1303	40.9748	28.7855	0.6	0.265
14.04.2012 15:24	468	481	627	0.0627	40.9748	28.7855	0.6	0.141
14.04.2012 15:30	468	481	1515	0.1515	40.9748	28.7855	0.6	0.302
14.04.2012 15:36	468	481	616	0.0616	40.9748	28.7855	0.8	0.139
14.04.2012 15:42	468	481	2181	0.2181	40.9748	28.7855	0.8	0.414
14.04.2012 15:48	468	481	2217	0.2217	40.9748	28.7855	1	0.420
14.04.2012 15:54	468	481	10,571	1.295	40.9748	28.7855	0.4	1.711
14.04.2012 16:00	468	481	3081	0.3081	40.9748	28.7855	0.6	0.562
14.04.2012 16:06	468	481	2164	0.2164	40.9748	28.7855	0.8	0.411
14.04.2012 16:12	468	481	2389	0.2389	40.9748	28.7855	0.6	0.449
14.04.2012 16:18	468	481	6165	0.6165	40.9748	28.7855	1	1.045
14.04.2012 16:24	468	481	4558	0.4558	40.9748	28.7855	1	0.796
14.04.2012 16:30	468	481	3813	0.3813	40.9748	28.7855	1.4	0.679
14.04.2012 16:36	468	481	4743	0.4743	40.9748	28.7855	1.2	0.825
14.04.2012 16:42	468	481	5964	0.5964	40.9748	28.7855	1.4	1.014
14.04.2012 16:48	468	481	2388	0.2388	40.9748	28.7855	2	0.449
14.04.2012 16:54	468	481	1705	0.1705	40.9748	28.7855	1.4	0.334
14.04.2012 17:00	468	481	8803	0.9414	40.9748	28.7855	1.4	1.446
14.04.2012 17:06	468	481	8179	0.8179	40.9748	28.7855	1	1.352
14.04.2012 17:12	468	481	6930	0.693	40.9748	28.7855	0.8	1.162
14.04.2012 17:18	468	481	6074	0.6074	40.9748	28.7855	0.8	1.031
14.04.2012 17:24	468	481	13,708	2.2064	40.9748	28.7855	2.6	2.177
14.04.2012 17:30	468	481	10,626	1.306	40.9748	28.7855	1.6	1.720
14.04.2012 17:36	468	481	5438	0.5438	40.9748	28.7855	2.2	0.933
14.04.2012 17:42	468	481	2695	0.2695	40.9748	28.7855	1	0.499
14.04.2012 17:48	468	481	7725	0.7725	40.9748	28.7855	0.8	1.283
14.04.2012 17:54	468	481	3536	0.3536	40.9748	28.7855	0.8	0.635
14.04.2012 18:00	468	481	6564	0.6564	40.9748	28.7855	0.6	1.107
14.04.2012 18:06	468	481	7914	0.7914	40.9748	28.7855	1	1.312
14.04.2012 18:12	468	481	8992	0.9792	40.9748	28.7855	0.8	1.475
14.04.2012 18:18	468	481	8796	0.94	40.9748	28.7855	1	1.445
14.04.2012 18:24	468	481	9975	1.1758	40.9748	28.7855	0.6	1.622
14.04.2012 18:30	468	481	6332	0.6332	40.9748	28.7855	0.4	1.071

(continued)

**Table 2.2** (continued)

Time (h)	Row	Column	Reflectivity	Calculated data (mm)	Longitude	Latitude	Ground data (mm)	Calculated PDF data (mm)	
14.04.2012 18:36	468	481	3072	0.3072	40.9748	28.7855	0.4	0.560	
14.04.2012 18:42	468	481	272	0.0272	40.9748	28.7855	0.4	0.070	
14.04.2012 18:48	468	481	567	0.0567	40.9748	28.7855	0.2	0.130	
14.04.2012 18:54	468	481	1139	0.1139	40.9748	28.7855	0.2	0.236	
14.04.2012 19:00	468	481	2081	0.2081	40.9748	28.7855	0.2	0.398	
14.04.2012 19:06	468	481	1608	0.1608	40.9748	28.7855	0.2	0.318	
14.04.2012 19:12	468	481	3372	0.3372	40.9748	28.7855	0.2	0.608	
14.04.2012 19:18	468	481	8828	0.9464	40.9748	28.7855	0.4	1.450	
14.04.2012 19:24	468	481	1999	0.1999	40.9748	28.7855	0.2	0.384	
14.04.2012 19:30	468	481	3700	0.37	40.9748	28.7855	0.2	0.661	
14.04.2012 19:36	468	481	4623	0.4623	40.9748	28.7855	0.2	0.806	
14.04.2012 19:42	468	481	3918	0.3918	40.9748	28.7855	0.2	0.695	
14.04.2012 19:48	468	481	2965	0.2965	40.9748	28.7855	0.6	0.543	
14.04.2012 19:54	468	481	2231	0.2231	40.9748	28.7855	0.4	0.423	
Arithmetic mean				0.495			Arithmetic mean	0.772	0.780
Standard deviation				0.415			Standard deviation	0.542	0.537

**Fig. 2.10** Gamma PDF: **a** ground rainfall measurements, **b** radar reflection measurements

same table also includes arithmetic mean and standard deviation values.

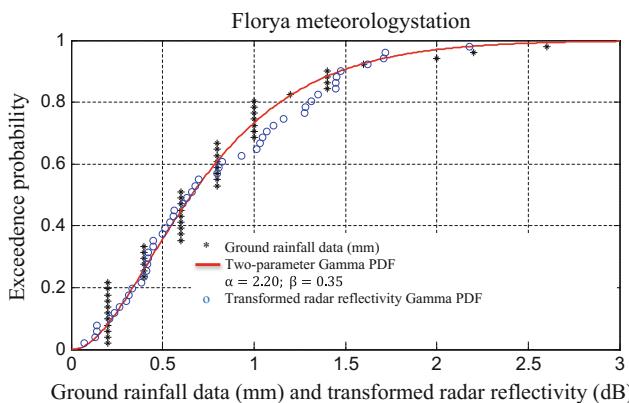
On the other hand, Fig. 2.11 presents actual ground measurements together with the radar reflection conversions

to rainfall on the same graph. It is clear that there is a very good agreement between two sets of values.

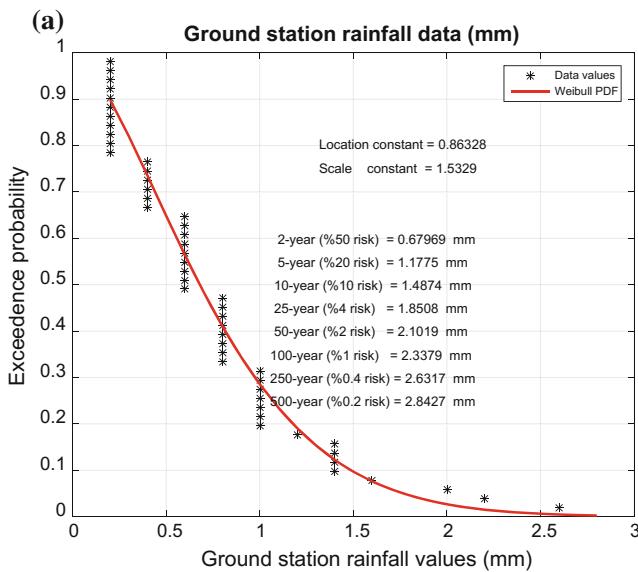
Figure 2.12 indicates the ground and radar reflection precipitation values' PDFs with a set of return periods

(1-year, 5-year, 10-year, 25-year, 50-year, 100-year, 250-year, and 500-year) and corresponding rainfall amounts. One can compare between the two rainfall amounts for each return period and appreciate that they are close to each other within the practically acceptable relative error limits.

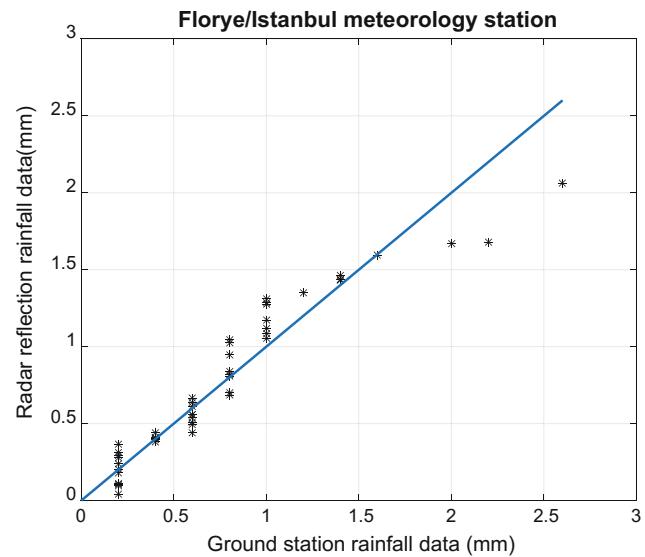
Finally, Fig. 2.13 indicates the plot of ground rainfall versus radar reflectivity-converted rainfall amounts, and one can observe that there are some significant deviations for high values only. However, this is the result of one single rainfall event, but in case of many rainfall records, the overall deviations might be rather very close to 1:1 ( $45^\circ$ ) straight-line.



**Fig. 2.11** Ground and radar reflectivity-converted amounts



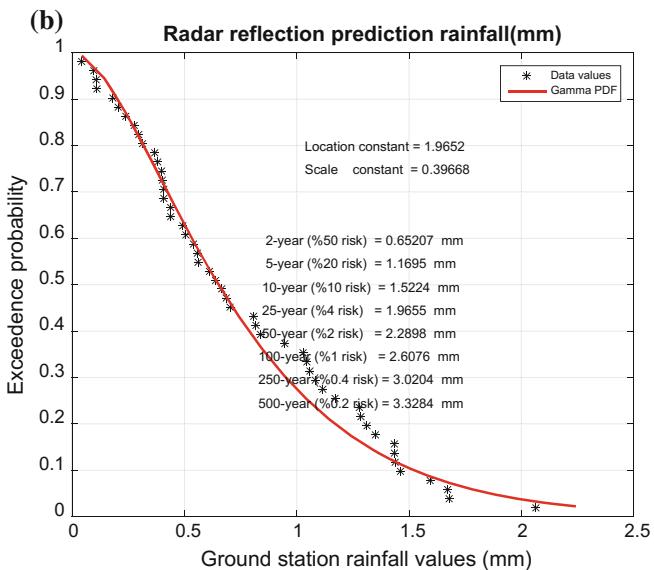
**Fig. 2.12** Rainfall amounts for return periods: **a** ground rainfall, **b** radar reflectivity rainfall



**Fig. 2.13** Scatter diagram between ground and radar rainfall amounts

## 2.5 Probable Maximum Precipitation (PMP)

The PMP calculation procedure can be found in the WHO manual (WMO 2009) provided that long data series (more than 15 years) are available. The basis of the statistical PMP calculation methodology is presented by considerations from the frequency equation by Hershfield (1965). In general, the



PMP is a function of the annual daily maximum precipitation arithmetic average,  $\bar{X}$ , and the standard deviation,  $S_X$ , of available daily maximum rainfall (ADMR) time series through a factor of the standard deviation as,

$$\text{PMP} = \bar{X} + k_m S_X \quad (2.14)$$

Herein,  $k_m$  is referred to as the “frequency factor” and it implies fraction of the standard deviations around the arithmetic mean to attain the largest possible precipitation value within a series. If the number of ADMR records is  $n$ , then the exclusion of the maximum one leads to similar

maximum daily rainfall series of length  $n - 1$  and new mean and standard deviation values similar to Eq. (2.14).

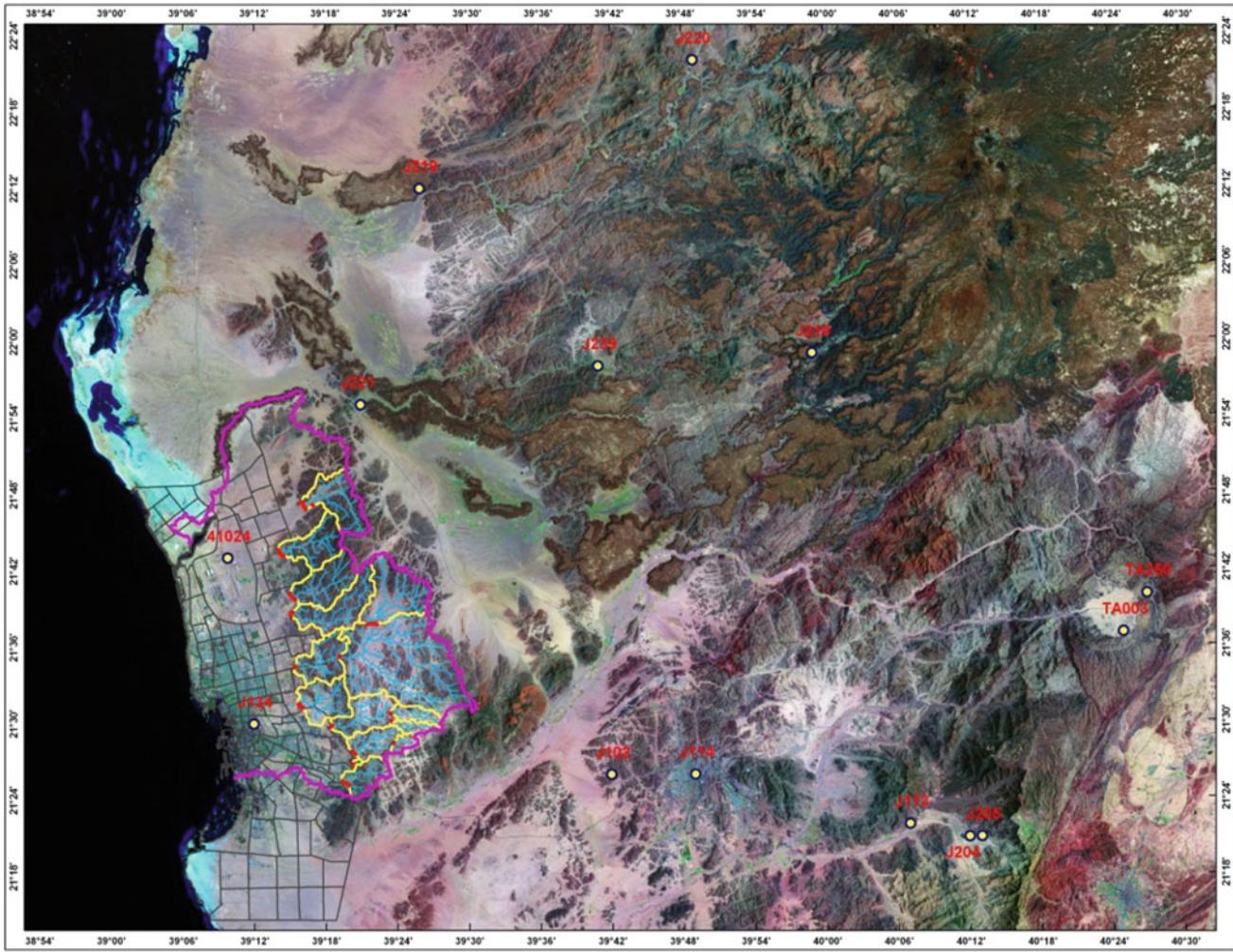
$$R_{\max} = \bar{X}_{n-1} + k_m S_{X(n-1)} \quad (2.15)$$

Hence, one can calculate the frequency factor as,

$$k_m = \frac{R_{\max} - \bar{X}_{n-1}}{S_{X(n-1)}} \quad (2.16)$$

The calculations can be achieved by the following software presented in MATLAB language.

```
function [m,s,m1,s1,k,PMP,r] = ProbableMaximumPrecipitation(X)
% This program has been written by Zekai Sen on 20 March 2014 for the
% SAUDI GEOLOGICAL SURVEY
% X Maximum Daily Rainfall
N=length(X); % The number of data
p=(1:1:N)./(N+1);
% Sort the data
SX=sort(X);
scatter(SX,1-p, 'k*')
hold on
% par=gamfit(SX);
par=gevfit(SX);
Xmax=max(X);
x=0:1:(Xmax+50);
% y=gamcdf(x,par(1),par(2));
y=gevcdf(x,par(1),par(2),par(3));
plot(x,1-y, 'k')
m=mean(SX(1:N));
s=std(SX(1:N));
m1=mean(SX(1:N-1));
s1=std(SX(1:N-1));
k=(Xmax-m1)/s1;
PMP=m+k*s;
xlabel('Maximum daily rainfall (mm)')
ylabel('Exceedence probability')
title('Probable Maximum Precipitation')
box on
text(50,0.9,['Gamma probability distribution function'])
text(50,0.8,['Alpha = ' num2str(par(1),3), ' mm', ' Beta = ' num2str(par(2),3), ' mm'])
text(50,0.7,['k factor = ' num2str(k,3), ' PMP = ' num2str(PMP,3), ' mm'])
text(100,0.5,[' * Data'])
text(100,0.4,[' - Theory'])
% Risk calculations, r, based on return periods, RP
RP(1)=2;
RP(2)=5;
RP(3)=10;
RP(4)=25;
RP(5)=50;
RP(6)=100;
RP(7)=250;
RP(8)=500;
for i=1:8
% r(i)=gaminv((1-1/RP(i)),par(1),par(2));
r(i)=gevinv((1-1/RP(i)),par(1),par(2),par(3));
end
end
```



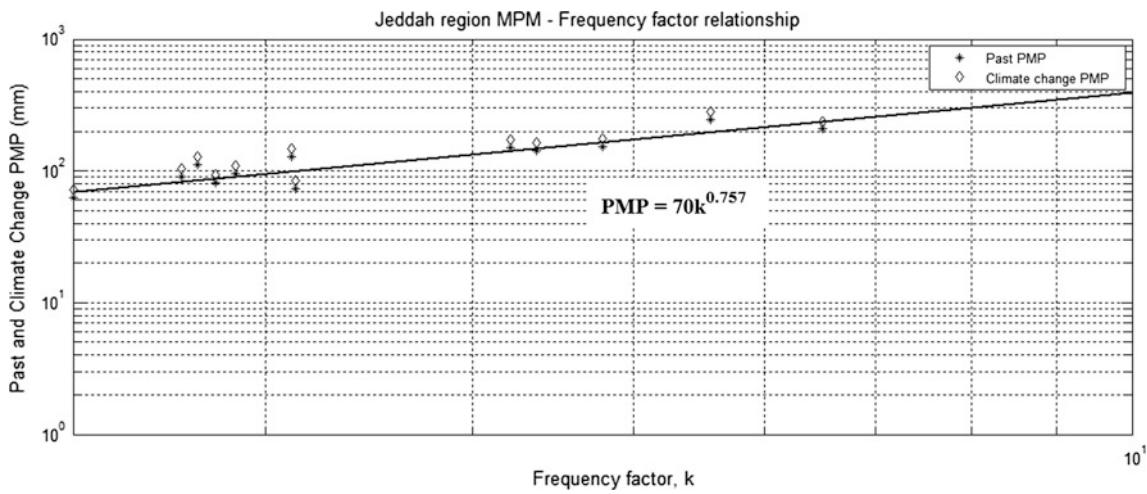
**Fig. 2.14** Meteorology station locations

The locations of meteorology stations are given in Fig. 2.14. In Table 2.3 (Sen 2014; Saudi Geological Survey Report), 12 meteorology station records are presented from

the Kingdom of Saudi Arabia for the PMP numerical calculations according to Eqs. (2.14)–(2.16).

**Table 2.3** PMP numerical values

Meteorology station	Record duration	Global parameters		Maximum exclusion parameters		Frequency factor	PMP (mm)	
		Mean (mm)	Standard deviation (mm)	Mean (mm)	Standard deviation (mm)		Past	Climate change
J102	1966–2011	28.40	20.70	26.30	16.06	4.22	151.00	173.65
J113	1966–2005	46.66	35.51	42.77	26.55	5.57	244.00	280.60
J114	1967–2013	36.96	27.44	33.69	23.86	2.73	112.06	128.87
J134	1970–2012	29.39	25.49	26.83	20.30	4.79	151.35	174.05
J204	1966–2005	36.50	20.60	35.10	18.91	2.88	95.87	110.25
J205	1966–2005	43.76	26.77	41.87	24.27	3.11	127.06	146.19
J214	1966–2006	32.20	25.10	30.02	20.99	4.37	141.99	163.29
J219	1970–2006	23.75	20.83	22.14	18.97	2.80	82.01	94.31
J220	1966–2011	21.67	16.55	20.57	15.08	3.13	73.50	84.53
J221	1971–2006	27.51	27.62	23.62	17.91	6.51	207.25	238.34
J239	1976–2011	24.38	16.37	23.24	15.34	2.30	62.10	71.41
41,024	1970–2009	28.94	22.74	27.25	20.89	2.67	89.61	103.05



**Fig. 2.15** PMP–frequency factor relationship

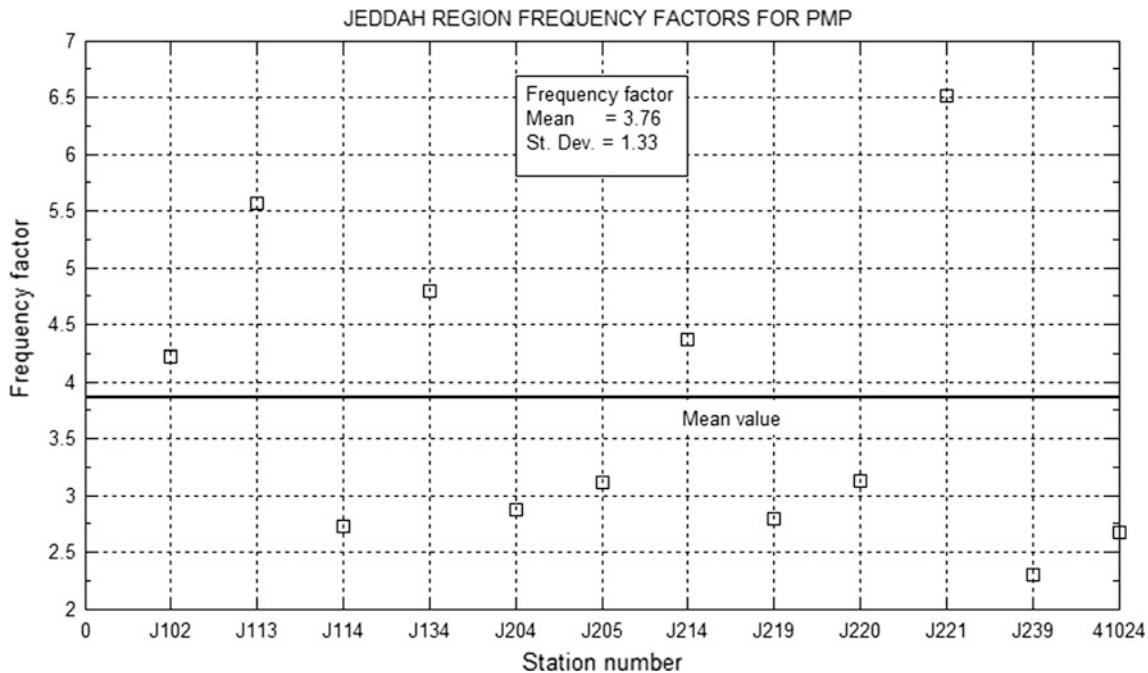
The application of Eq. (2.16) with the arithmetic mean and standard deviation values in columns 5 and 6 yields the frequency factor values in column 7. Table 2.3 also indicates that the frequency factor values vary between 2.30 and 6.51. On the other hand, one can observe from the same table that one-day past PMP is the biggest (244 mm) at station J113. The same is valid for the climate change effective calculations. There is a logarithmic relationship between the PMP and its corresponding frequency factor as in Fig. 2.15.

This logarithmic relationship in the figure between the PMP and the frequency factor is given as follows.

$$PMP = 70k^{0.757} \quad (2.17)$$

Figure 2.16 presents the frequency factor distribution with the meteorological station. The arithmetic mean and the standard deviation of the frequency factor are 3.76 and 1.33, respectively.

The maximum and minimum frequency factors appear at the J221 and J239 meteorology stations, respectively. For the regional assessment, one can use the arithmetic average of the frequency factor, which is equal to 3.76; however, such a work is not necessary, because the regionalization is worked with the daily maximum rainfall amounts



**Fig. 2.16** Frequency factor distributions with meteorology stations

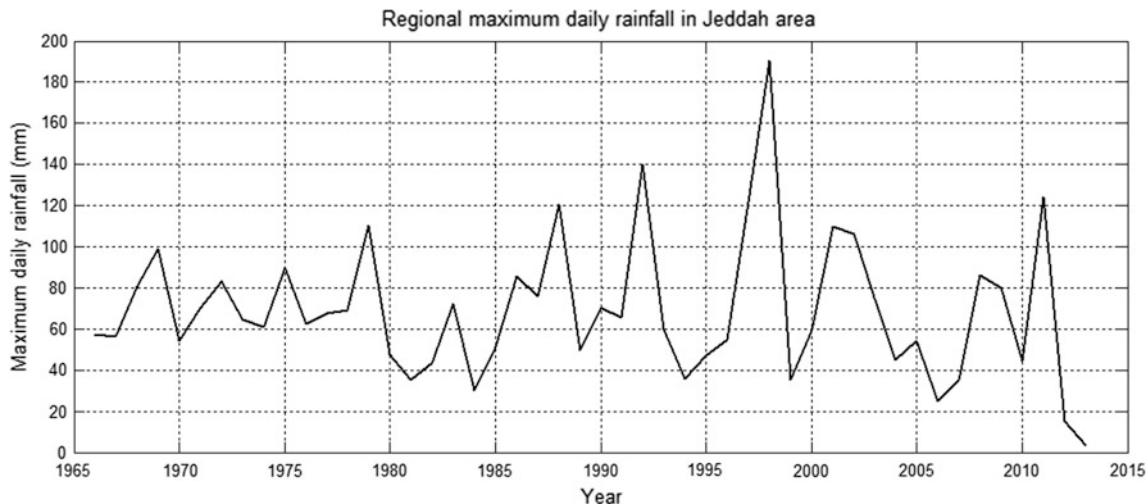
irrespective of the meteorology stations as shown in the last column in Table 2.3; Fig. 2.17 indicates the change of annual maximum daily rainfall pattern.

The daily annual maximum rainfall time series has the Gamma PDF with the relevant parameters (Fig. 2.18).

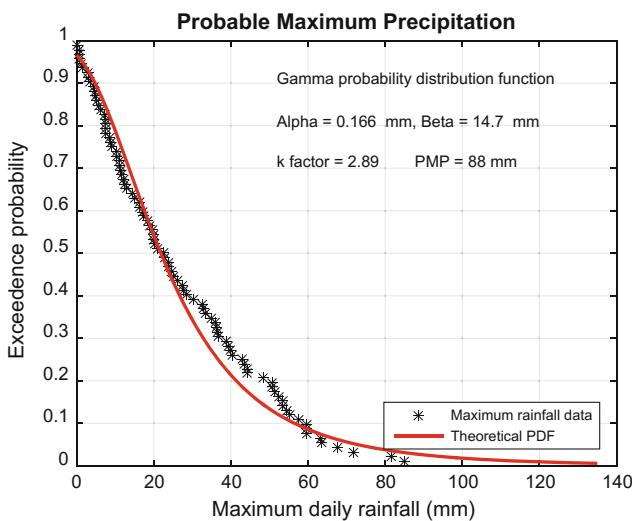
The PMP (213 mm) is the greater than any other station. The MATLAB program is given in the following box as,

```
function [m,s,m1,s1,k,PMP,r] = ProbableMaximumPrecipitation(X)
```

```
function [m,s,m1,s1,k,PMP,r] = ProbableMaximumPrecipitation(X)
% This program has been written by Zekai Sen on 20 March 2014 for the
% X      : Maximum daily rainfall record
% m      : The arithmetic average of the whole time series
% s      : The standard deviation of the whole time series
% m1     : The arithmetic average of the time series with exclusion of the
%          last record
% s1     : The standard deviation of the time series with exclusion of the
%          last record
% k      : The frequency factor
% PMP    : Probably maximum precipitation value
% r      : The risk levels
N=length(X); % The number of data
p=(1:1:N)./(N+1);
% Sort the data
SX=sort(X);
figure
scatter(SX,1-p,'k*')
hold on
par=gevfit(SX);
Xmax=max(X);
x=0:1:(Xmax+50);
y=gevcdf(x,par(1),par(2),par(3));
plot(x,1-y,'r','LineWidth',2)
m=mean(SX(1:N));
s=std(SX(1:N));
m1=mean(SX(1:N-1));
s1=std(SX(1:N-1));
k=(Xmax-m1)/s1;
PMP=m+k*s;
xlabel('Maximum daily rainfall (mm)')
ylabel('Exceedence probability')
title('Probable Maximum Precipitation')
box on
text(50,0.9,'Gamma probability distribution function')
text(50,0.8,['Alpha = ' num2str(par(1),3), ' mm', ' Beta = ' num2str(par(2),3), ' mm'])
text(50,0.7,['k factor = ' num2str(k,3), ' PMP = ' num2str(PMP,3), ' mm'])
legend('Maximum rainfall data','Theoretical PDF','Location','SouthEast')
box on
grid on
% Risk calculations, r, based on return periods, RP as follows
RP(1)=2;
RP(2)=5;
RP(3)=10;
RP(4)=25;
RP(5)=50;
RP(6)=100;
RP(7)=250;
RP(8)=500;
for i=1:8
    r(i)=gevinv((1-1/RP(i)),par(1),par(2),par(3)); % Risk calculations
end
end
```



**Fig. 2.17** Annual maximum daily rainfall time series



**Fig. 2.18** Annual maximum daily rainfall characteristics

Figure 2.19 shows the change of PMP along the meteorology stations for both past records and climate change projections.

This relationship can be regarded as the regional expression for the PMP calculation in the Jeddah City region, Kingdom of Saudi Arabia. Out of 12 stations, seven (J102, J113, J114, J134, J205, J214, and J221) have PMP values more than 100 mm in case of past record bases. Climate change effect increases this number to nine stations including J204 and 41024. Relatively low maximum daily rainfall amounts, especially those less than 100 mm, are observed at meteorology stations, J220 and J239. On the other hand, storm cells are observed at the time, which

resulted in extremely high rainfall levels at stations other than those included in the ongoing analysis. The long-term annual average rainfall for Jeddah City is approximately 55 mm. It is estimated that some 160 mm of rainfall fell on November 25, 2009, for about 3-h duration leading to the disastrous consequences. It is obvious from Table 2.3 that the first three PMP records have been observed at J113 (190.6 mm), J221 (140.2 mm), and J205 (117.4 mm). The maximum rainfall amounts are related to the movement of moist westerly or northwesterly fronts or large areas of low pressures. Additionally, another potential reason is due to the flow of hot and humid tropical air from the Red Sea Region and its convergence with polar marine air over the study area.

In order to see how close are the actually recorded daily maximum rainfall amounts to the calculated PMP values, the efficiency factor,  $E$ , is defined as the ratio of the maximum daily rainfall,  $R_{\max}$ , to the calculated past and climate effective PMPs.

$$E = \frac{\text{PMP}}{R_{\max}} \quad (2.18)$$

Sixth and seventh rows in Table 2.4 are the efficiency values, which are always more than one. It implies that in the study region the PMP values are not yet reached by the recorded rainfall occurrences, and hence, there is still room for more dangerous rainfall occurrences. At places the difference is more than 70%, which implies that for future designs, concerning any engineering structure (dams, diversion canals, roads, etc.), more care must be taken into consideration and to be on the safe side the structures must be constructed based on the PMP values.

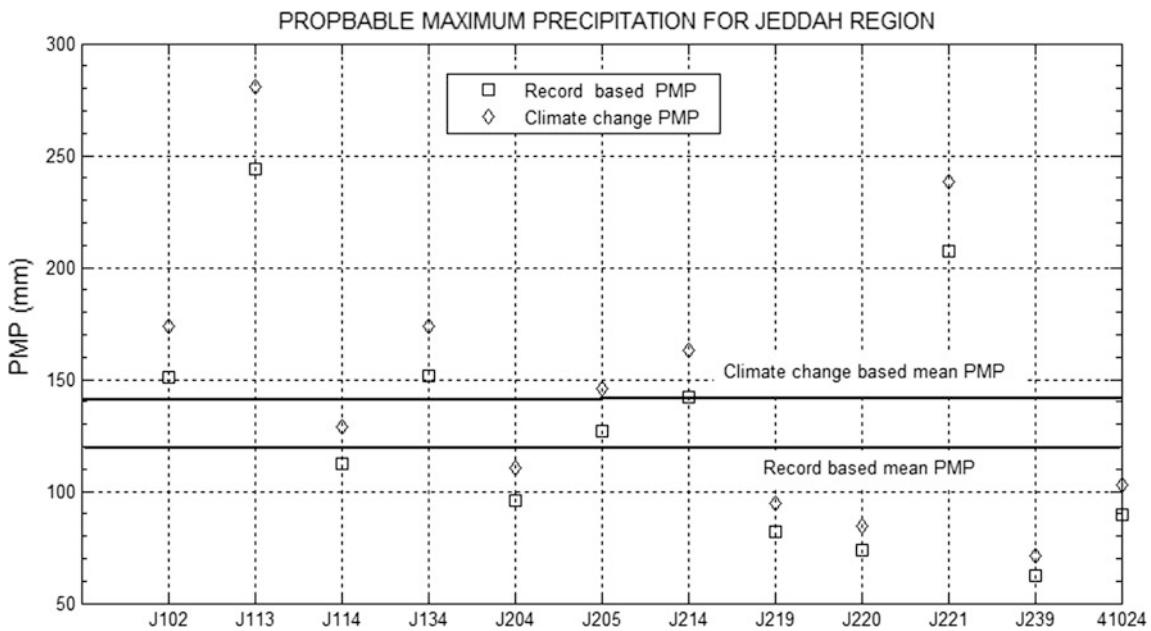


Fig. 2.19 PMP change along stations

Table 2.4 Rainfall efficiency

Years	Stations											
	J102	J113	J114	J134	J204	J205	J214	J219	J220	J221	J239	41,024
$R_{\max}$	110.5	190.6	99	124	89.6	117.4	121.8	75.2	67.8	140.2	58.6	83
PMP (past)	151	224	112.1	151.4	95.9	127.1	142	82.01	73.5	207.3	62.1	89.6
PMP (climate)	173.7	280.6	128.9	174.1	110.3	146.2	163.3	94.3	84.6	238.3	71.4	103.1
Efficiency (past)	1.37	1.18	1.13	1.22	1.07	1.08	1.17	1.09	1.08	1.48	1.06	1.08
Efficiency (climate)	1.57	1.47	1.30	1.40	1.23	1.25	1.34	1.25	1.24	1.70	1.22	1.24

## 2.6 Meteorological Water Balance

Thorntwaite (1948) has suggested monthly water balance calculations, and for this purpose, annual periods are selected. Especially, monthly records of temperature and precipitation are taken into consideration. As a simple water balance principle actual and potential evapotranspiration rates, soil moisture storage and its capacity, direct flow, snow storage and melt rates, and surplus flow amounts are calculated. The structural elements of such a model are presented in Fig. 2.20.

The first estimations from the water balance are for monthly precipitation, rainfall, and snow in mm. If the temperature,  $T$ , is below a threshold value,  $T_{\text{snow}}$ , then precipitation is totally in the form of snow. If the temperature is above the rainfall threshold level,  $T_{\text{rain}}$ , the precipitation is completely in rainfall type. However, in case of  $T_{\text{snow}} < T < T_{\text{rain}}$  the amount of snow decreases linearly from 100 to 0% according to the following expression. In this range, the snow is referred to as watery snow.

$$S = P \left( \frac{T_{\text{rain}} - T}{T_{\text{rain}} - T_{\text{snow}}} \right) \quad (2.19)$$

On the other hand, the rainfall amount can be calculated as,

$$R = P - S \quad (2.20)$$

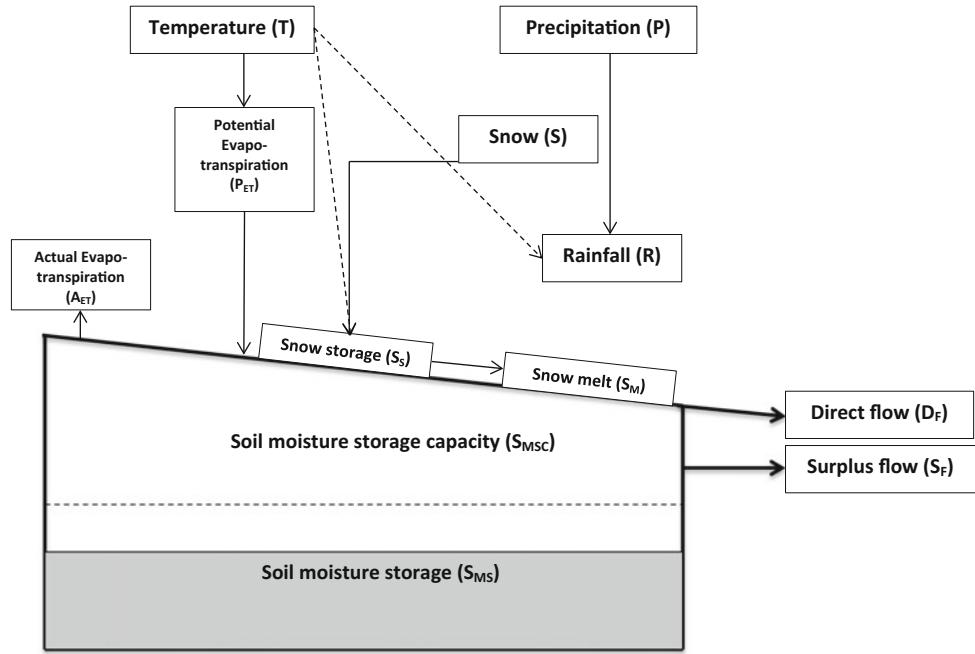
In many preliminary works,  $T_{\text{rain}}$  value is considered as 3 °C; for elevations less than 1000 m  $T = -10$  °C, and for elevations more than 1000 m  $T = -1$  °C (Thorntwaite 1948).

For the calculation of direct flow,  $D_F$ , a certain amount of rainfall is considered with the proportionality coefficient,  $\alpha$ , which is expressed in percentage, and for practical applications, it is recommended as %5.

$$D_F = \alpha P \quad (2.21)$$

The remaining precipitation,  $P_R$ , is the difference between the precipitation and direct flow.

$$P_R = P - D_F \quad (2.22)$$

**Fig. 2.20** Water balance model

The snow melt ratio,  $S_{MR}$ , can be obtained by considering average monthly temperature and the melting ratio,  $\beta$ , which is recommended as 0.5 (Wolock and McCabe 1999).

$$S_{MR} = \beta \left( \frac{T - T_{\text{snow}}}{T_{\text{rain}} - T_{\text{snow}}} \right) \quad (2.23)$$

If  $S_{MR} > \beta$ , then  $S_{MR} = \beta$ . Monthly snow melt,  $S_M$ , can be estimated as,

$$S_M = S \times S_{MR} \quad (2.23)$$

In addition to the remaining precipitation,  $P_R$ , the snow melt,  $S_M$ , total water entering the soil,  $W_{TS}$ , can be calculated.

Actual evapotranspiration,  $A_{ET}$ , potential evapotranspiration,  $P_{ET}$ , soil moisture storage,  $S_{MS}$ , and soil moisture storage withdraw,  $S_{MW}$ , are calculated. Monthly  $P_{ET}$ , is calculated from the average monthly temperature in such a way that it is defined as always moist and vegetation cover causes water loss.  $P_{ET}$  is the amount of water requested by the climate. Its calculation can be achieved by formulation (Hamon 1961),

$$P_{ET} = 13.97ds^2W_D \quad (2.24)$$

where  $d$  is the number of days in one month,  $s$  is the average daily sunshine duration hour within 12 h, and  $W_D$  is the saturated water vapor density in  $\text{gr/m}^3$ . On the other hand,  $W_D$  can be expressed in terms of temperature as,

$$W_D = \frac{4.95e^{0.062T}}{100} \quad (2.25)$$

In this chapter, another  $P_{ET}$  calculation method has been adapted based on the latitudinal position, NL, of the meteorology station and monthly temperature,  $T_i$  ( $i = 1, 2, \dots, 12$ ), along the following formulation list of first the monthly Thornthwaite heat index,  $I_i$ , related to the mean monthly temperature as,

$$I_i = \left( \frac{T_i}{5} \right)^{1.514} \quad (2.26)$$

As the next stage the annual heat index,  $H$ , is calculated in terms of the monthly heat indices,  $I_i$  as,

$$H = \sum_{i=1}^{12} I_i \quad (2.27)$$

The  $P_{ET}$  calculation for each month,  $i$ , with theoretical sunshine hours per day is given by Thronthwaite (1948) as follows.

$$P_{ETi} = 16 \left( \frac{10T_i}{H} \right)^\alpha \quad (2.28)$$

where the power parameter is given empirically as,

$$\alpha = 675 \times 10^{-9}H^3 - 771 \times 10^{-7}H^2 + 1792 \times 10^{-5}H + 492 \times 10^{-3} \quad (2.29)$$

If in any month  $W_{TS} < P_{ET}$ , then  $A_{ET} = (W_{TS} + S_{WD})$ . If  $S_{WD}$  decreases in a linear form with  $S_{MS}$ , then as the soil becomes dry, water withdrawal provides difficulty, and hence, there remains less water for  $A_{ET}$ . For  $S_{WD}$  calculation in terms of the previous month  $S_{MS}$  ( $S_{MS}$ )<sub>i</sub> the following expression is valid.

$$S_{WD} = (S_{MS})_{i-1} - \left\{ |P - P_{ET}| \left[ \frac{(S_{MS})_{i-1}}{S_{MSC}} \right] \right\} \quad (2.30)$$

McCabe and Wolock (1999) indicated that in many places  $S_{MSC}$  can be taken as 150 mm. If  $(P \text{ or } Z_{WD}) < P_{ET}$ , then water deficit is  $W_D = P_{ET} - A_{ET}$ . On the other hand, if  $P > P_{ET}$ , then  $A_{ET} = P_{ET}$  and water that is bigger than  $P_{ET}$  feeds the soil,  $S_{MS}$ . Finally, if  $S_{MS} > S_{MSC}$  then water surplus appears as surface flow.

Monthly direct flow,  $D_F$ , generation is a certain percentage of water surplus, and in general, 50% is employed for calculations.

All the calculation steps in this section are put into a MATLAB program form with the main and subprograms, respectively, as,

```
function [Prain,Psnow,Ptotal,PET,AET,WD,RO] = MeteorologicalWaterBalance(P,T,Pt,St,STi,STC,IY,NL,Title)
function [PET] = ThornthwaitePET(T,IY,NL,Title)
```

These two functions in MATLAB programming language are given in the following two boxes, and the input and output variables are explained in detail in the programs.

```
function [Prain,Psnow,Ptotal,PET,AET,WD,RO] = MeteorologicalWaterBalance(P,T,Pt,St,STi,STC,IY,NL,Title)

% This program is written by Zekai Sen on 25 October 2014 at 23:57 p.m.
% IMPORTANT NOTICE: THE FIRST SEQUENCE IS DEFINED BY A SINGLE VARIABLE IN THE PROGRAM,
% FOR INSTANCE SUCH AS D(:,1),D(:,2),D(:,3),D(4:,4)

% INPUTS
% P      : Monthly precipitation time series
% T      : Monthly temperature time series
% PET    : Thornthwaite potential evapotranspiration empirical calculation
%          based on the monthly temperature values
% Pt     : Rainfall start temperature: 3 oC
% St     : Snow temperature IF > 1,000 m THEN - 10 oC
%          IF < 1,000 m THEN - 1 oC
% STi    : Initial soil-moisture storage
% STC    : Soil-moisture capacity: 150 mm
% OUTPUTS
% AET    : Actual evapotranspiration
% Ptotal: Total rainfall
% WD    : Water deficit
% SP    : Surplus
% STi   : Soil-moisture storage
% STC   : Soil-moisture capacity
% STW   : Soil-moisture storage withdrawal
% ROD   : Direct runoff
% RO    : Total Runoff
% IY    : Initial year
% NL    : North latitude
[PET] = ThornthwaitePET(T,IY,NL,Title);
Name=({'Precipitation (mm)', 'Snow (mm)', 'Total water input (mm)', ...
        'Potential evapotranspiration (mm)', 'Actual evapotranspiration (mm)', ...
        'Water deficit (mm)', 'Surface flow (mm)'});

n=length(P);
Y=1:1:n;% Y Year sequence
% Snow accumulation calculation
for i=1:n
    if T(i) > Pt
        Prain(i)=P(i); % Rainfall
        Psnow(i)=0.0;
    else
        if T(i)< Pt && T(i) > St
            Psnow(i)=P(i)*(Pt-T(i))/(Pt-St); % Snow accumulation storage
            Prain(i)=P(i)-Psnow(i); % Rainfall
        else
            Psnow(i)=P(i);
            Prain(i)=0;
        end
    end
end
..... continued .....
```

```
% Plot rainfall and snow graphs
figure
plot(Y,Prain,'b','LineWidth',2)
title([Title, ' Rainfall']); xlabel('Years'); ylabel(Name(1)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
figure
plot(Y,Psnow,'b','LineWidth',2)
title([Title, ' Watery snow']); xlabel('Years'); ylabel(Name(2)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
% Direct runoff calculation
Drf=0.35; % Direct runoff fraction: THIS CAN BE CHANGED ACCORDING TO THE SITUATION
for i=1:n
    ROD(i)=Prain(i)*Drf; % Direct runoff
    Premain(i)=Prain(i)-ROD(i); % Rainfall losses for infiltration+evapotranspiration
end
% Snow melt calculation
Mmr=0.5; % Maximum melt ratio
for i=1:n
    if T(i) > St && T(i)< Pt
        Smeltf(i)=Mmr*(T(i)-St)/(Pt-St);
        Smelt(i)=Psnow(i)*Smeltf(i); % Snow melt water equivalent
        Ptotal(i)=Smelt(i)+Premain(i); % Total loss water, which does not occur as runoff
    else
        Smelt(i)=0;
        Ptotal(i)=Premain(i);
    end
end
% Total water plot
figure
plot(Y,Ptotal,'b','LineWidth',2)
title([Title, ' Total water input']); xlabel('Years'); ylabel(Name(3)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
% Evaporation and soil moisture calculations
ny=n/12; % Number of years
ST(1)=STi; % Initial soil-moisture storage
for i=1:n
    if Ptotal(i) < PET(2,i)
        AET(i)=Ptotal(i)+ST(1);
    else
        AET(i)=PET(2,i);
    end
end
% Potential evapotranspiration plot
%figure
%plot(Y,PET,'b','LineWidth',2)
%title([Title, 'Potential evapotranspiration']); xlabel('Years'); ylabel(Name(4)); grid on; box on
%Soil-moisture storage withdrawal calculation
%xticks([0:240:1200])
%xticklabels({'2001','2020','2040','2060','2080','2100'})
```

..... continued .....

```

for i=2:n
    if Ptotal(i) < PET(2,i)
        STW(i)=ST(i-1)-(PET(2,i)-Ptotal(i))*(ST(i-1)/STC);
        AET(i)=Ptotal(i)+STW(i);
        ST(i)=ST(i-1)-STW(i);
    else
        AET(i)=PET(2,i);
        ST(i)=ST(i-1)+(Ptotal(i)-PET(2,i));
        STW(i)=0.0;
    end
    if Ptotal(i)+STW(i) < PET(2,i)
        %
        WD(i)=PET(i)-AET(i); % Water deficit
        WD(i)=PET(2,i)-AET(i); % Water deficit
    else
        WD(i)=0;
    end
    if ST(i) > STC
        SP(i)=ST(i)-STC; % Surplus calculation
        RC=0.5*SP(i); % Runoff contribution from surplus
        RO(i)=RC+ROD(i);
        ST(i)=STC;
    else
        RO(i)=ROD(i);
    end
end
% Actual evapotranspiration plot
for i=1:n
    if AET(i) < 0
        AET(i)=10;
    else
    end
end
figure
plot(Y,AET,'b','LineWidth',2)
title([Title, ' Actual evapotranspiration']); xlabel('Years'); ylabel(Name(5)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
% Water deficit plot
figure
plot(Y,WD,'b','LineWidth',2)
title([Title, ' Water deficit']); xlabel('Years'); ylabel(Name(6)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
% Runoff plot
figure
plot(Y,RO,'b','LineWidth',2)
title([Title, ' Surface flow']); xlabel('Years'); ylabel(Name(7)); grid on; box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
end

```

```

function [PET] = ThornthwaitePET(T,IY,NL>Title)
% This program is written by Zekai Sen on 16 July 2018
% T      : Monthly air temperature in Celsius 1x12 values
% IY     : Initial year
% NL    : North latitude (0 < NL <60)
N=length(T);          % Number of months
NY=floor(N/12);       % Number of years
K=[0      1      1      1      1      1      1      1      1      1      1      1
10    0.97    0.98    1    1.03    1.05    1.06    1.05    1.04    1.02    0.99    0.97    0.96
20    0.92    0.98    1    1.05    1.09    1.11    1.1    1.07    1.02    0.98    0.93    0.91
30    0.87    0.98    1    1.07    1.14    1.17    1.16    1.11    1.03    0.96    0.89    0.85
40    0.8    0.89    0.99    1.1    1.2    1.25    1.23    1.15    1.04    0.93    0.83    0.78
50    0.71    0.84    0.98    1.14    1.28    1.36    1.33    1.21    1.06    0.9    0.76    0.68
60    0.54    0.67    0.97    1.19    1.33    1.56    1.55    1.33    1.07    0.84    0.58    0.48
]; % K coefficient in the Thornthwaite PET calculation depending on north latitude and months
for m=1:NY
    INY=IY;           % Initial year
    FNY=INY+NY;       % Final year
    YR=INY:1:FNY;    % Whole years
    TT=T(12*(m-1)+1:12*m);
    I=(TT/5).^1.514;
    J=sum(I);
    c=0.000000675*J^3-0.0000771*J^2+0.01792*J+0.49239;
    MNL=K(:,1)/NL; % Middle value of latitude
    for i=1:
        if MNL(i,1) == 1
            PETT=16*K(i,2:13).*(10*TT'./J).^c;
        else
            end
    end
    for i=1:6
        if MNL(i) < 1 && MNL(i+1) > 1
            LNL=K(i,1);
            UNL=K(i+1,1);
            R=(NL-LNL)/(UNL-LNL);
            CK=K(i,2:13)+R*(K(i+1,2:13)-K(i,2:13)); % Calculation K
            PETT=16*CK(1,1:12).*(10*TT/J).^c;
        else
            end
    end
    PETM(12*(m-1)+1:12*m)=PETT; % Monthly PET
end
k=0;
NM=12*NY;
for i=1:NM
    k=k+1;
    PET(k)=PETM(1,i);
end
for i=1:N
    if PET(i) < 0
        PET(i)=0;
    else
        end
end
YM=1:1:N; % Number of months
PET=[YM;PET];
figure
plot(PET(1,:),PET(2,:),'b','LineWidth',2)
title>Title
xlabel'Years'
ylabel'Potential evapotranspiration (mm)'
grid on
box on
xticks([0:240:1200])
xticklabels({'2001','2020','2040','2060','2080','2100'})
end

```

The results from the software are presented in the following table for each water balance component for only 3 years (Table 2.5).

The graphical outputs from the above main and subprograms are given in Fig. 2.21 for each water component that is estimated from the meteorology water balance methodology.

**Table 2.5** Water balance results

Year	Month	Rainfall (mm)	Snow (mm)	Total rainfall (mm)	PET (mm)	AET (mm)	Water deficit (mm)	Surface runoff (mm)
2001	1	58,738	34,588	49,064	0	0	0	0
	2	158,118	129,802	138,419	0	0	0	55,341
	3	640,630	0	416,410	188,193	188,193	0	224,220
	4	979,980	0	636,987	509,481	509,481	0	342,993
	5	710,400	0	461,760	756,915	1,260,272	0	248,640
	6	338,220	0	219,843	1,020,933	310,996	709,937	118,377
	7	49,576	0	32,224	1,570,933	29,529	1,541,405	17,352
	8	151,580	0	98,527	1,197,454	127,184	1,070,270	53,053
	9	98,508	0	64,030	945,824	96,390	849,434	34,478
	10	582,250	0	378,462	549,810	419,345	130,464	203,788
	11	144,930	0	94,205	277,699	98,831	178,868	50,725
	12	521,790	0	339,164	55,631	55,631	0	182,627
2002	1	121,781	0.8489	83,126	40,021	40,021	0	42,623
	2	162,332	64,578	128,615	0	0	0	56,816
	3	601,740	0	391,131	283,709	283,709	0	210,609
	4	133,820	0	86,983	372,056	543,243	0	46,837
	5	68,815	0	44,730	816,857	96,680	720,177	24,085
	6	0.6618	0	0.4302	1,025,392	21,896	1,003,495	0.2316
	7	426,940	0	277,511	1,435,838	286,056	1,149,782	149,429
	8	60,347	0	39,226	1,345,608	42,965	1,302,643	21,121
	9	550,330	0	357,715	765,533	376,085	389,448	192,615
	10	482,160	0	313,404	566,123	319,108	247,015	168,756
	11	46,667	0	30,334	303,139	31,279	271,860	16,333
	12	136,488	20,272	97,543	22,142	22,142	0	47,771
2003	1	405,580	0	263,627	67,398	67,398	0	141,953
	2	123,718	0.8072	84,205	47,726	47,726	0	43,301
	3	852,030	0	553,820	218,480	218,480	0	298,210
	4	324,380	0	210,847	307,400	813,075	0	113,533
	5	780,460	0	507,299	831,380	539,779	291,601	273,161
	6	260,460	0	169,299	1,171,127	172,272	998,855	91,161
	7	48,822	0	31,734	1,454,227	32,043	1,422,183	17,088
	8	68,219	0	44,342	1,372,039	44,994	1,327,045	23,877
	9	185,440	0	120,536	795,578	123,296	672,282	64,904
	10	345,740	0	224,731	452,289	226,647	225,642	121,009
	11	232,080	0	150,852	216,826	151,180	65,647	81,228
	12	39,056	0.8013	28,711	13,502	13,502	0	13,669

The following MATLAB program converts a given monthly time series into Thornthwaite convenient format.

manner similar to Fig. 3.5 in Chap. 3, one can identify a series of dry (drought) periods. The time scale can be considered as 3 or 6 months for short and 12, 24, or 48 months

```

function [D] = ArrangeDataForThornthwaite(X,IY,Du)
% This program arranges the row data to convenient Thornthwaite
% program data, which requires the data in the first, second and the third
% column as year, month and data.
% X is the data matrix with rows as years and the columns include 12
% monthly values
% IY is the initial year as 2015
% Du is the duration as 5 years,
k=0;
Y=IY-1;
for i=1:Du
    Y=Y+1;
    for j=2:13
        k=k+1;
        D(k,1)=Y;
        D(k,2)=j-1;
        D(k,3)=X(i,j);
    end
end
end

```

## 2.7 Standard Precipitation Index (SPI)

SPI is designed for rainfall reduction determination at different time scales (McKee et al. 1993, 1995). A given time series  $X_i$  ( $i = 1, 2, \dots, n$ ) with arithmetic average,  $\bar{X}$  and standard deviation,  $S_X$ , is transformed into a standard series,  $x_i$ , as,

$$x_i = \frac{X_i - \bar{X}}{S_X} \quad (2.31)$$

Such a standardization operator provides the following significant properties about the standard time series.

- (1) The arithmetic average,  $\bar{x}_i$ , of the standardized time series is equal to zero,
- (2) The standard deviation of the standardized series time series is equal to one,
- (3) This standard time series has rather small positive and negative deviations from zero level, where negative values are among drought indicator,
- (4) The standardized series has no dimension,
- (5) The standard series is expected to abide by the normal (Gaussian) probability distribution function (PDF); if not then the values should be converted to normal PDF by some transformation.

Drought period starts with the first negative SPI value along the series and ends with the first positive value. In this

for longer durations, where more dry spells start to emerge. Drought classifications are given in Table 2.6.

The application of SPI method has to adapt the following steps in sequence for 1-month, 3-month, 6-month, 12-month, 24-month, and 48-month basic durations. In these steps, MATLAB statements are also given conveniently (Sen 2009a).

- (1) The CDF of the given time series is obtained, and the graph is plotted for visual inspection,
- (2) Especially for time scales less than 12-month, the Gamma PDF seems as the most frequently valid one,
- (3) If the data at hand is labeled shortly as “**data**”, then for the Gamma PDF the MATLAB command is **gamfit** (**data**), which provides location,  $\alpha$ , and scale,  $\beta$  parameters,
- (4) In order to match the most suitable PDF with the parameters, the theoretical probability values are calculated by **y = gamcdf(x, α, β)** MATLAB statement,
- (5) The theoretical PDF can be plotted onto the data scatter graph by **hold on** command and then **plot (x, y)** statement,
- (6) The Gamma PDF must now be converted to a standard normal (Gaussian) PDF. For this purpose, the probabilities for each data value must be calculated from a Gamma PDF by MATLAB statement **p = gamedf(-data, α, β)**,
- (7) These probability values are then entered into the standard normal (Gaussian) PDF function so as to calculate the corresponding data values by statement

- SPI = norminv(p, 0, 1).** Hence, the standard data values that correspond to a normal PDF are ready for interpretations,
- (8) In order to check the standardization of the last data set, one can plot SPI values versus probabilities by **plot(SPI, p)** MATLAB statement. Hence, one can visualize the normal PDF,
- (9) Finally, the change of standard SPI values are plotted against time by **plot(SPI)** MATLAB command, and hence, the SPI graph is ready for interpretations.

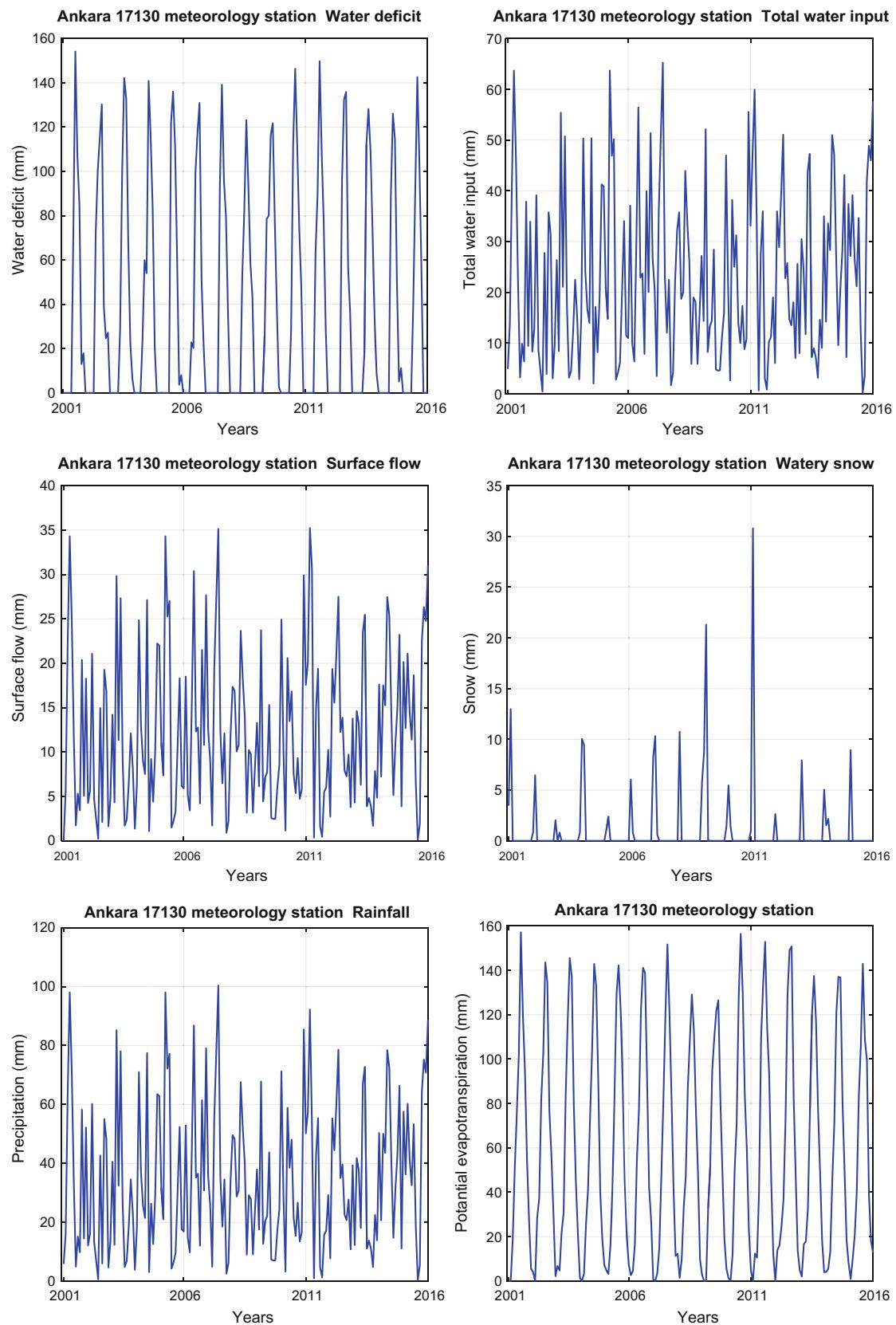
All the aforementioned steps can be repeated for any duration (3-month, 6-month, 12-month, 24-month, 48-month).

The following MATLAB program converts a Gamma PDF to a normal (Gaussian) PDF for use in the SPI calculation.

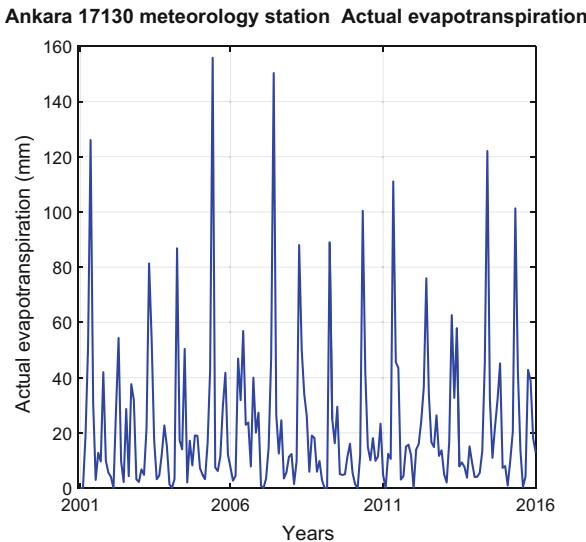
```

function [UV,UZS,GZS] = DistributionTransformation(X,Title,UYs)
% This program is written by Zekai Sen on 23 July 2014
% X = Two dimensional monthly data set(Yearx12)
% NPD = Standard distribution converted data
% UZS = Time series generation (Yearx12 length)
% GZS = Time series observation (Yearx12) length
% UV = Final monthly generation data (Year row x 12 column)
% UYS = Number of generation year
% pr = Gamma probability distribution parameters
% X=X(X>0);
ny=length(X(:,1));
for i=1:12
    XP=X(1:ny,i);
    % XPP=XP(XP>0);
    XPP=XP+0.00001;
    np=length(XPP);
    p=(1:1:np)/(np+1);
    XS=sort(XPP);
    scatter(XS,p,'k*')
    hold on
    pr(i,1:2)=gamfit(XPP);
% The probability distribution function (PDF) selection for the data and % probability calculations
    Prob(1:np,i)=gamcdf(XPP,pr(i,1),pr(i,2));
%    Prob(1:np,i)=wblcdf(XPP,pr(i,1),pr(i,2));
% Standard normal PDF value calculations that corresponds to probabilities
    NPD(1:np,i)=norminv(Prob(1:np,i),0,1);
    NPD(np+1:ny,i)=0;
    Xm(i)=min(XPP)+0.00001;
    XM(i)=max(XPP);
    XMM(i)=XM(i)+0.15*XM(i);
    x=Xm(i):1:XMM(i);
    y=gamcdf(x,pr(i,1),pr(i,2));
%    y=wblcdf(x,pr(i,1),pr(i,2));
    plot(x,y)
    xlabel('Title')
    ylabel('Non-exceedence probability')
    figure
end
% Generation of standard normal probability distribution
[GX] = HuzmeUretimi(NPD,Title,UYS);
% Generation of generated variables that accord with the Gamma PDF
for i=1:12
    ProbNor=normcdf(GX(1:ny,i),0,1);
    UV(1:ny,i)=gaminv(ProbNor,pr(i,1),pr(i,2));
%    UV(1:ny,i)=wblinv(ProbNor,pr(i,1),pr(i,2));
end
% Sequence generation
k=0;
for j=1:ny
    for i=1:12
        k=k+1;
        UZS(k)=UV(j,i);
        GZS(k)=X(j,i);
    end
end
end

```



**Fig. 2.21** Various components of meteorological water balance



**Fig. 2.21** (continued)

The outputs of the above step-by-step procedure are presented in Fig. 2.22 for Istanbul location, Turkey. Herein, only three samples are presented; if the reader wants to increase it for any number of monthly periods, aforementioned steps are sufficient for their generation.

## 2.8 Harmonic Analysis

As already mentioned earlier each precipitation time series,  $P_i$  ( $i = 1, 2, \dots, n$ ) may include possibly five parts as (Şen 2009b),

- (1) Linear increasing or decreasing trend,  $T_i$  (Chap. 10),
- (2) Seasonality,  $S_i$ , in the form of regular fluctuations,
- (3) Sudden change as shifts or jump,  $J_j$  ( $J_i = 0$  except for  $i = j$ ),
- (4) Stochastic component,  $X_i$  (Chap. 6),
- (5) Completely random part, i.e., residuals (random errors),  $\varepsilon_i$  (Chap. 6).

Seasonality component exists in records less than annual duration (season, month, week, and hour) time series as in Fig. 2.23.

Prior to the application of harmonic analysis to any time series, it is recommended to convert it to a series with zero mean. After the detrending operation (Chap. 10), the seasonality can be expressed as the summation of a certain number,  $m$ , of sine and cosine waves with different phases as in the following formulation.

**Table 2.6** SPI drought classifications

SPI value	Classes
0.00 to (0.99)	Light
(-1.00) to (-1.49)	Medium
(-1.50) to (-1.99)	Intensive
<(-2.00)	Extreme

$$X_i = \sum_{i=1}^m \left[ A_j \sin\left(2\pi \frac{j}{n} i\right) + B_j \cos\left(2\pi \frac{j}{n} i\right) \right] \quad (2.32)$$

Since the summation of sine and cosine waves has zero mean, this expression results in zero arithmetic average value, but the summation of various waves leads to the global amplitude, which represents the standard deviation of the zero mean time series. However, in case of nonzero arithmetic average,  $\bar{X} \neq 0$ , time series formulation takes the following form.

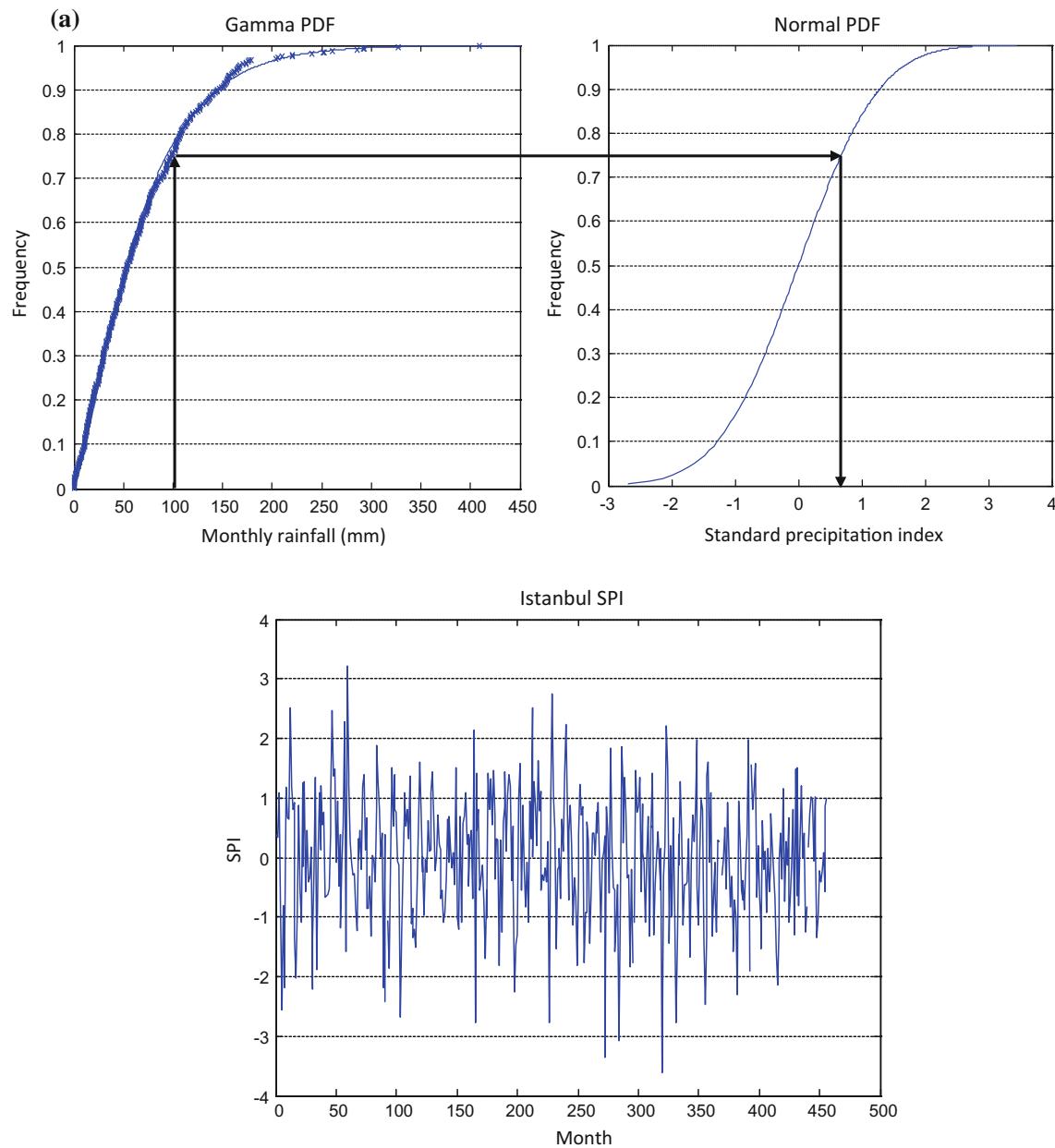
$$X_i = \bar{X} + \sum_{i=1}^m \left[ A_i \sin\left(2\pi \frac{j}{n} i\right) + B_i \cos\left(2\pi \frac{j}{n} i\right) \right] \quad (2.33)$$

As the summation number increases, the harmonic time series approaches to the original time series closer. It is also possible to represent the original time series exactly without any error if summation counter,  $m$ , is equal to the number of data,  $n$ . However, in practical applications, most often a certain amount of error percentage,  $\varepsilon_i$ , is accepted, and in this case the last equation can be rewritten with error component as,

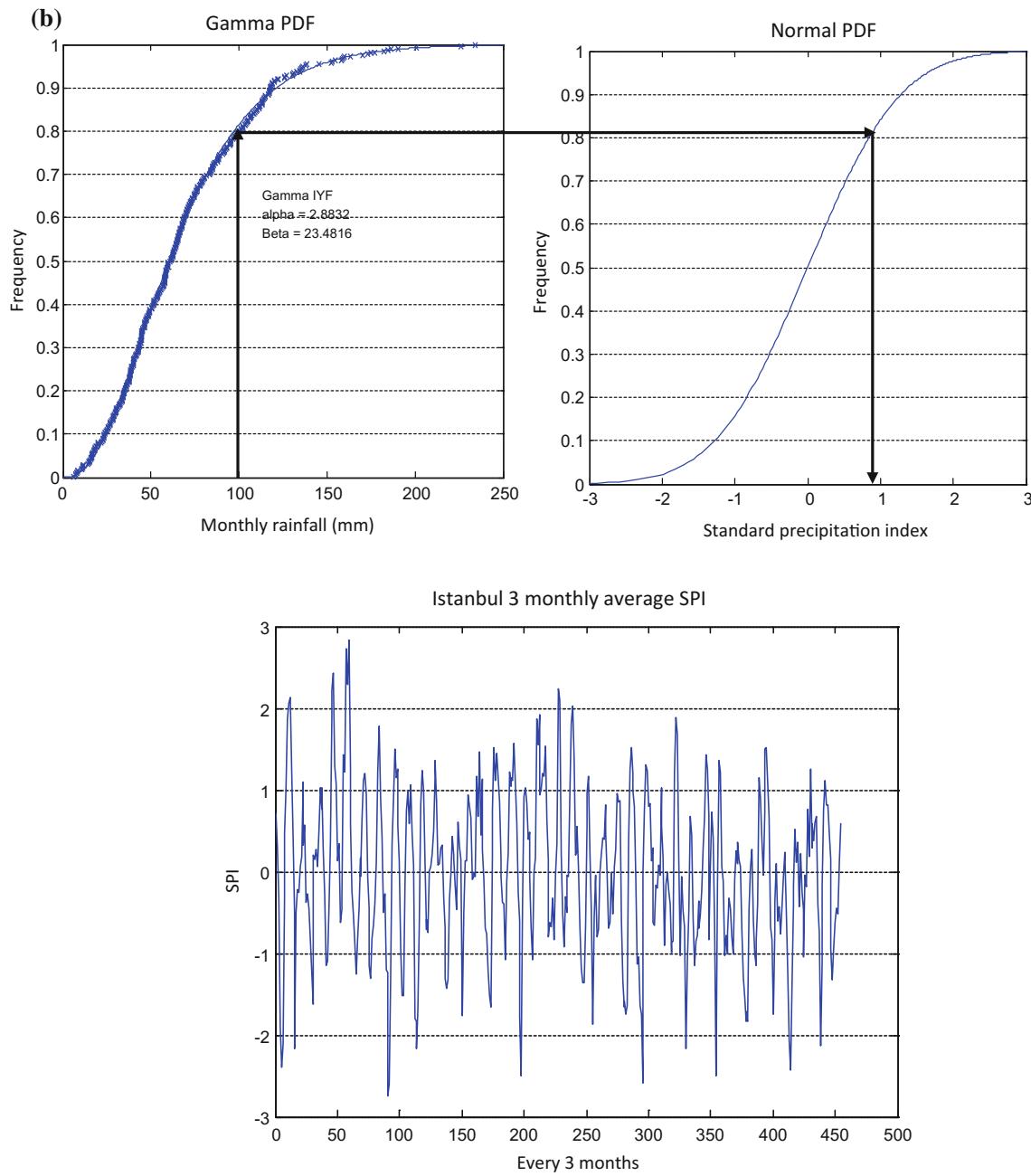
$$X_i = \bar{X} + \sum_{i=1}^m \left[ A_i \sin\left(2\pi \frac{j}{n} i\right) + B_i \cos\left(2\pi \frac{j}{n} i\right) \right] + h_i \quad (2.34)$$

The sine and cosine waves decompose the original time series into its regular components with  $1/n, 2/n, \dots, m/n$  harmonics. In the expression,  $A_j$ 's and  $B_j$ 's ( $j = 1, 2, \dots, m$ ), are  $2m$  unknown parameters; they must be estimated from the available time series data. For this purpose, the least squares error procedure is considered. In Eq. (2.34) after making the error term object, taking the square of both sides and the global summation of these terms; the partial derivation of  $\text{Min} (\sum i^2)$  with respect to  $A_j$  and  $B_j$  and then their equality to zero lead the parameter estimations as,

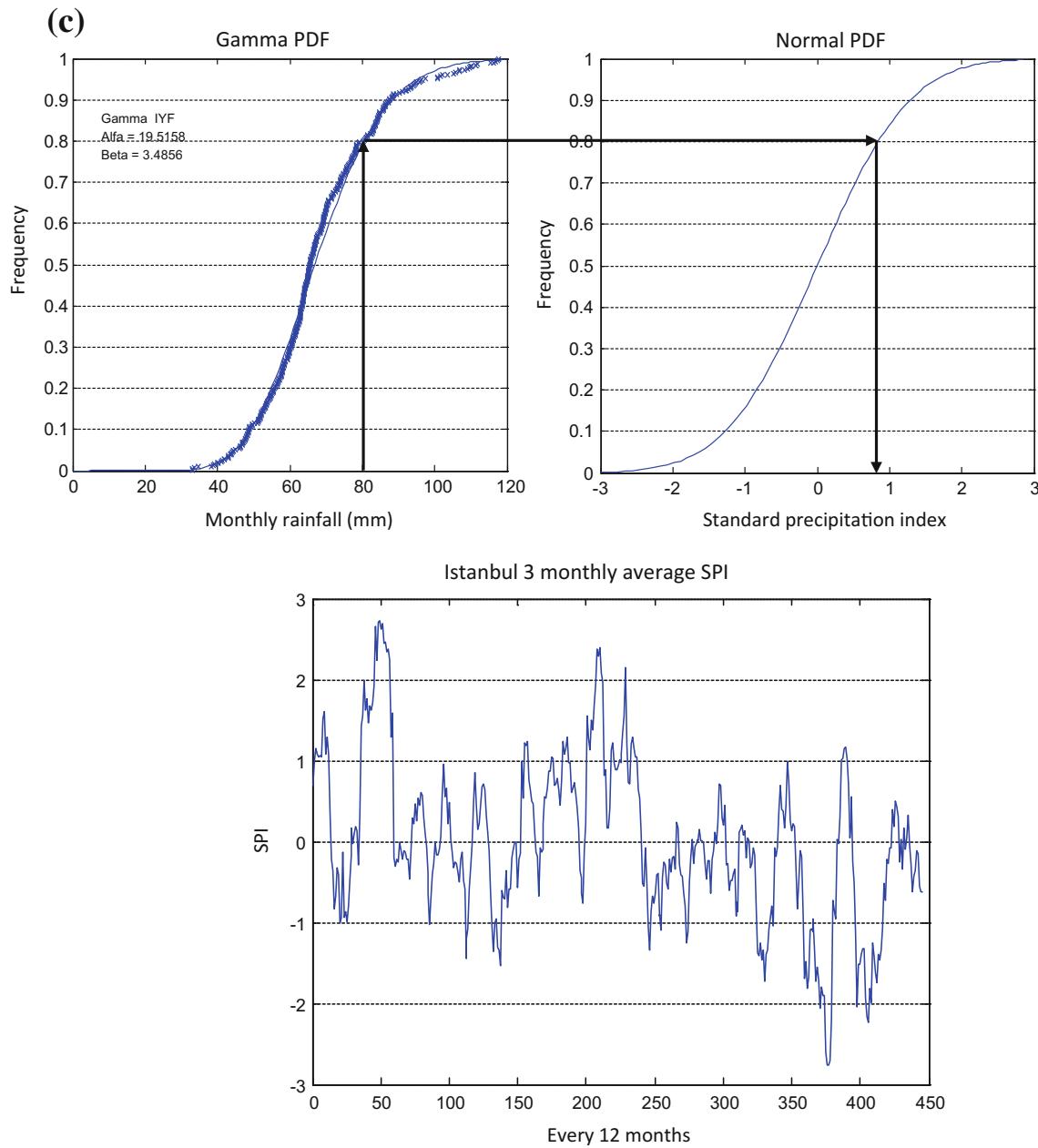
$$A_j = \frac{2}{n} \sum_{i=1}^{n-1} X_i \cos\left(2\pi \frac{j}{n} i\right) \quad (2.35)$$



**Fig. 2.22** PDF and SPI time variations for observation and theoretical Gamma PDF data: **a** 1-month duration, **b** 3-month duration, **c** 12-month duration



**Fig. 2.22** (continued)

**Fig. 2.22** (continued)

and

$$V_j = A_j^2 + B_j^2 \quad (2.37)$$

$$B_j = \frac{2}{n} \sum_{i=1}^{n-1} X_i \sin\left(2\pi \frac{j}{n} i\right) \quad (2.36)$$

The right-hand sides of these last two equations are known in terms of available time series. The variance,  $V_j$ , and the phase angle,  $\Theta_j$  of the harmonic time series can be obtained as follows.

And the phase angle as,

$$\Theta_j = \tan^{-1}\left(\frac{A_j}{B_j}\right) \quad (2.38)$$

The MATLAB program for all the above calculations is given in the following box.

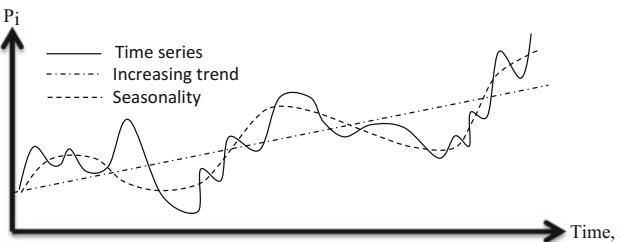
```

function [xh,a1,b1] = HarmonicAnalysis(X,m,T,TX,TY)
% X      : Time series data
% Y      : Time interval series such as minute, day, month, year, etc.
% m      : The number of harmonics
% T      : The title of the data type
% TX     : X axis title with units. For instance 'Time (Year)' and alike
% TY     : Y axis title with units. For instance 'Height (m)' and alike
n=length(X);          % Number of data
Y=1:1:n;              % Number of x axis values
xmean=mean(X);        % Arithmetic average of given data
xs=X-xmean;           % Data series with zero arithmetic average
c=2/n;
for i=1:n
    a=0;
    b=0;
    for j=1:n
        a=a+xs(j)*cos(2*pi*i*j/n);
        b=b+xs(j)*sin(2*pi*i*j/n);
    end
    a1(i)=c*a; % Sine terms coefficients
    b1(i)=c*b; % Cosine terms coefficients
end
for j=1:n
    s=0;
    for i=1:m
        s=s+a1(i)*cos(2*pi*i*j/n)+b1(i)*sin(2*pi*i*j/n);
    end
    xf(j)=s; % Harmonic function with zero arithmetic average
end
xh=mean(X)+xf; % Harmonic function with actual arithmetic average
figure
plot(Y,X)
title(T)
hold on
plot(Y,xh,'r')
xlabel(TX)
ylabel(TY)
grid on
box on
legend('Location','Northeast','Measurement data','Harmonic analysis data')
end

```

### 2.8.1 Known Seasonality Case

In cases of known seasonality, such as monthly or daily time series, instead of the above-mentioned harmonic analysis, the seasonality in the arithmetic average and the standard deviation can be removed by considering the seasonal arithmetic averages and standard deviations. If for  $n$  days



**Fig. 2.23** Different components of a time series

hourly records are available, then there are  $N = 24n$  data values. Such a time series element can be shown in a sequence as  $y_0, y_1, y_2, \dots, y_{N-1}$ , and it can be arranged as in Table 2.7.

In the last two rows, each column has the arithmetic average, ( $\bar{Y}_i, i = 0, 1, 2, \dots, 23$ ), and standard deviation ( $S_i, i = 0, 1, 2, \dots, 23$ ) values. Subtraction of the column arithmetic average from corresponding column values in Table 2.7 leads to Table 2.8, where the periodicity in the arithmetic average has been eliminated.

In this table each column has zero arithmetic average, but the periodicity remains in the standard deviation. In order to eliminate the periodicity in the standard deviation, each column value is divided by the corresponding standard deviation and, as a result, the new time series becomes as in Table 2.9.

**Table 2.7** Hourly periodic data

$y_0$	$y_1$	$y_2$	—	—	—	—	—	$y_{23}$
$y_{24}$	$y_{25}$	$y_{26}$	—	—	—	—	—	$y_{47}$
$y_{48}$	$y_{49}$	$y_{50}$	—	—	—	—	—	$y_{60}$
—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—
$y_{24(n-1)}$	$y_{24(n-1)+1}$		—	$y_{24(n-1)+2}$	—	—	—	$y_{24n-1}$
$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_3$	—	—	—	—	—	$\bar{Y}_{23}$
$S_0$	$S_1$	$S_2$	—	—	—	—	—	$S_{23}$

**Table 2.8** Time series free of periodicity in the arithmetic average

$y_0 - \bar{Y}_0$	$y_1 - \bar{Y}_1$	—	—	—	—	$y_{23} - \bar{Y}_{23}$
$y_{24} - \bar{Y}_0$	$y_{25} - \bar{Y}_1$	—	—	—	—	$y_{47} - \bar{Y}_{23}$
$y_{48} - \bar{Y}_0$	$y_{49} - \bar{Y}_1$	—	—	—	—	$y_{60} - \bar{Y}_{23}$
—	—	—	—	—	—	—
—	—	—	—	—	—	—
$y_{24(n-1)} - \bar{Y}_0$	$y_{24(n-1)+1} - \bar{Y}_1$	—	—	—	—	$y_{24n-1} - \bar{Y}_{23}$
$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	—	—	—	$\mathbf{0}$
$S_0$	$S_1$	—	—	—	—	$S_{23}$

**Table 2.9** Standardized time series

$(y_0 - \bar{Y}_0)/S_0$	$(y_1 - \bar{Y}_1)/S_1$	—	—	—	—	$(y_{23} - \bar{Y}_{23})/S_{23}$
$(y_{24} - \bar{Y}_0)/S_0$	$(y_{25} - \bar{Y}_1)/S_1$	—	—	—	—	$(y_{47} - \bar{Y}_{23})/S_{23}$
—	—	—	—	—	—	—
—	—	—	—	—	—	—
$(y_{24(n-1)} - \bar{Y}_0)/S_0$	$(y_{24(n-1)+1} - \bar{Y}_1)/S_1$	—	—	—	—	$(y_{24n-1} - \bar{Y}_{23})/S_{23}$
$\mathbf{0}$	$\mathbf{0}$	—	—	—	—	$\mathbf{0}$
$\mathbf{1}$	$\mathbf{1}$	—	—	—	—	$\mathbf{1}$

Finally, the values in this table are free of periodicity in the arithmetic average and the standard deviation. Known

periodicity case seasonality elimination can be achieved by the following MATLAB program.

```

function [Y,AX,SX,AY,SY] = StandardMonthlyTimeSeries(X)
% This program standardizes a given monthly time series into a standardized
% monthly time series, where the monthly arithmetic averages are equal to
% zero and monthly standard deviations is equal to 1. However there
% remains seasonality (monthly variations) in the serial correlation
% functions only.
% X is the monthly data [(n-year row)x(12-month column)]
% Y is the standardized monthly time series [(n-year row)x(12-month column)]
N=length(X(:,1)); % number of years
% Calculation of the monthly arithmetic averages and standard deviations
for i=2:13
    AX(i)=mean(X(:,i)); % Time series monthly arithmetic averages
    SX(i)=std(X(:,i)); % Time series monthly standard deviations
end
% Calculate the standardized monthly series
for i=1:12
    for j=1:N
        Y(j,i)=(X(j,i)-Ave(i))/Std(i); % Standardized series
    end
end
% Calculation of the monthly arithmetic averages and standard deviations of
% standardized series
for i=2:13
    AY(i)=mean(Y(:,i)); % Standardized time series monthly arithmetic averages
    SY(i)=std(Y(:,i)); % Standardized time series monthly standard deviations
end
end

```

## References

- Chow, V.T. 1964. Handbook Of Applied Hydrology. McGraw- Hill, New York.
- Cornell, J.R., and Benjamin, C.A., (1970). Probability, Statistics, and Decisions for Civil Engineers. Dover Books and Engineering: 677 pp.
- Gumbel, E.J. 1958. Statistics Of Extremes, Columbia University Press, New York, NY.
- Hamon, W.R., (1961). Estimating potential evapotranspiration: Journal of the Hydraulics Division, Proceedings of the American Society of Civil Engineers, Vol. 87: 107–120.
- Hazen, A. 1914. Storage To Be Provided In Impounding Reservoirs For Municipal Water Supply, Trans. ASCE, 77, 1308, ASCE, New York, NY.
- Hershfield, D. M., (1965).Method for estimating probable maximum rainfall. Journal (American Water Works Association): 965–972.
- Kite, G.W. 1977. Frequency and Risk Analysis in Hydrology, Water Res. Publications, Fort Collins, CO.
- McCabe, G.J., and Wolock, D.M., (1999). Future snowpack conditions in the western United States derived from general circulation model climate simulations: Journal of the American Water Resources Association, Vol. 35: 1,473–1,484.
- McKee, T. B., Doesken, N.J., and Kleist, J., (1993). The relationship of drought frequency and duration to time scales. In: Preprints, Eighth Conference on Applied Climatology (Anaheim, California): 179–184.
- McKee, T. B., Doesken, N. J., and Kleist, J., (1995). Drought monitoring with multiple time scales. In: Preprints, Ninth Conf. on Applied Climatology (Dallas, Texas): 233–236.
- Pearson, K. 1930. Tables of Statisticians And Biometricalians, Part I, The Biometric Laboratory, University College, London; printed by Cambridge University Press, London, 3rd ed.
- Şen, Z., (2008). Wadi Hydrology. Taylor and Francis Group, CRC Press, Boca Raton: 247 pp.
- Şen, Z., (2009a). Kuraklık Afet ve Modern Hesaplama Yöntenleri. (Drought Hazard and Modern Calculation Methods) Su Vakfı Publications: 248 pp. (in Turkish).
- Şen, Z., (2009b). Spatial Modeling Principles in Earth Science. Springer: 351 pp.
- Şen, Z., (2014). Probable maximum precipitation (PMP) and probable maximum flood (PMF). Saudi Geological Survey (SGS): 65 pp.
- Şen, Z., (2015). Intensity-duration-frequency (IDF) curve generations for Saudi Arabian meteorology stations. Saudi Geological Survey (SGS). Open Report: 76 pages.
- Şen, Z., (2017). Flood Modeling, Prediction, and Mitigation. Springer Verlag, 422 pp.
- Thornthwaite, C.W., (1948). An approach toward a rational classification of climate. The Geog ~ Rtv. 38(1): 55–94.
- WHO, World Meteorological Organization, (2009). Manual on Estimation of Probable Maximum Precipitation (PMP) WMO-No. 1045, 257 pp.
- Wolock, D.M., and McCabe, G.J., (1999). Effects of potential climatic change on annual runoff in the conterminous United States: Journal of the American Water Resources Association, Vol. 35: 1,341–1,350.
- Yevjevich, V. 1972. Probability and Statistics in Hydrology, Water Resources Publications, Fort Collins, CO.

## 3.1 General

The root meaning of the word hydrology is water science that deals with the occurrence, movement, and distribution of water in the earth with a search of its physical quantities and chemical qualities. In practice, it is concerned with water cycle major elements including evaporation, precipitation, runoff, and groundwater recharge. Although there are different aspects in hydrology research domain, the most important topics are flood and drought features and their predictions to hinder harmful consequences on human society. There are theoretical and practical solution methodologies among which the rational and empirical alternatives are used most often to get simple but effective results. Hydrological calculations are rather complex procedures, because the surface water and groundwater recharge events are dependent not only on the precipitation amounts, but also on a set of different causative factors such as the surface features (geomorphology), land use, geology, topography, antecedent precipitation conditions and also on the engineering structures performance within the study area. The studies are based mainly on the drainage basin, which is defined as the area that collects rainfall through the runoff toward the lowest elevation outlet point. Water quantity measure is discharge, which is the volume of water per time interval. In many engineering projects, the peak (maximum) discharge is the most important design value according to which different applications (dam, reservoir, bridge, culvers, channel, etc.) are completed under a certain risk acceptance (Chap. 2). In addition to the peak discharge, the evolution of runoff discharge along the time axis is also sought and the curve that shows the change of discharge with time after a storm rainfall is referred to as a hydrograph with its three parts, namely the rising limb coupled with the storm rainfall transformation depending on the geology and morphology of drainage basin, the recession limb after the cessation of storm rainfall, and the peak discharge location in between

the limbs. The volume underneath a hydrograph yields surface water volume that results from a single storm rainfall. Hydrographs are very useful especially for flood routing, which provides the change of flood wave during its movement toward the lower elevations through a natural or manmade channel.

The main purpose of this chapter is to present flood and drought calculation methodologies with unit hydrograph and hydrograph analyses. The necessary MATLAB language written programs are among the supportive gadgets for effective studies.

## 3.2 Digital Elevation Model (DEM)

DEM provides digital representation of any portion of Earth terrain in three-dimension (3D) leading to basic topographic and morphologic quantities about a drainage basin. They are available in the electronic medium at different resolutions and, therefore, very convenient to use through software that serve rapid and effective solutions. DEM resolution depends on scale and the data source (digital satellite images, aerial photographs) and the spatial resolution (i.e., grid spacing) and on variables like data structure and algorithms that are used during the extraction process. Remote sensing data has a similar structure to DEM except that each pixel shows brightness values. DEM, remote sensing, and satellite data in electronic media help to make manipulation, classification, analysis, and interpretation in imaginary world in computers (Campbell 2002). In general, DEMs provide the following direct information or help to derive some other beneficial information.

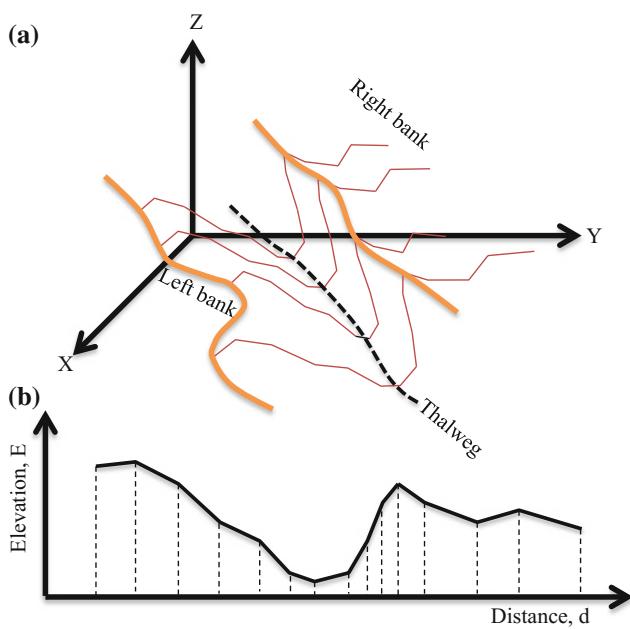
- (1) Elevation of the earth's surface points and area,
- (2) Properties of the gradient vector; the magnitude defines slope, and the direction angle represents terrain aspect,
- (3) Surface curvature,

- (4) Convexity,  
 (5) Surface-specific points and lines, i.e., local maxima (peaks), minima (pits), saddle points (passes), inflection points, break-lines, ridges, and valley lines. Simple representations of 3D and 2D (cross-sections) earth surface features are shown in Fig. 3.1.

In order to be able to depict the flood inundation boundaries, it is necessary to know the topographic map at least along the water channel or nowadays one can obtain the electronic version on computers through the DEM. However, even though the DEM data are available, it is recommended that one should make few field trips to refine the DEM results and compare them with actual positions. The simplest form to determine the inundation area is by the bank-full levels consideration at a set of cross-sections along the channel concerned.

### 3.2.1 Gradients (Slopes)

It is possible to calculate slope electronically along any direction from the availability of DEM data. Hence, 3D DEM surface can be converted to 3D drainage basin slope map in the form of aspects. In practical hydrology applications, most often either the average slope of a drainage basin or the main channel or cross-section slopes are important for



**Fig. 3.1** a 3D view, b 2D cross-section

engineering calculations. The slope of drainage basin is its vertical drop per unit of horizontal distance, and it plays a dominant role in runoff, flood velocity, and discharge calculations. In the upstream drainage basin high-elevation areas, the slope is the steepest, but the steepness reduces gradually toward the outlet point. It is possible to appreciate the slope along the wadi from upstream toward the downstream along the longitudinal profile of the main channel, which has generally a concave shape. In high elevations, the surface water erodes deep valleys in the hilly and mountainous terrain due to the high runoff velocity. The steeper the slope, the more rapidly flows the runoff. Therefore, the time to peak is shorter and the peak discharge is higher in the upstream portions than downstream. On the other hand, infiltration capacity, and hence, the groundwater recharge is meager at steep slope locations (Sen 2017).

Although there are different slope measures, the simplest slope,  $S$ , is defined as the ratio of difference between the elevations ( $E_1$  and  $E_2$ ) to the distance,  $D$ , between two points.

$$S = \frac{E_2 - E_1}{D} \quad (3.1)$$

### 3.2.2 Cross-Sections

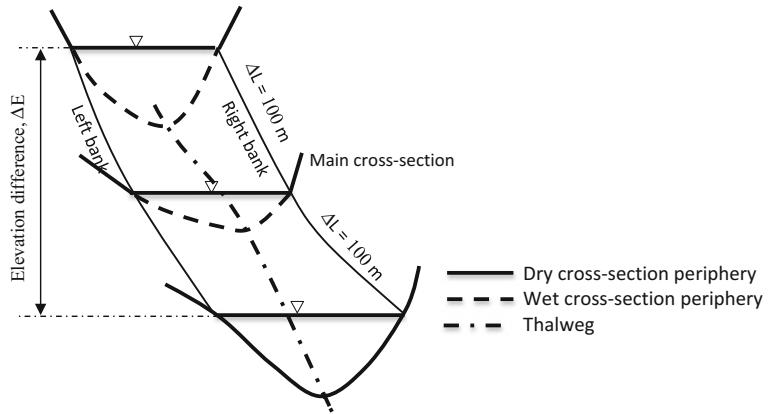
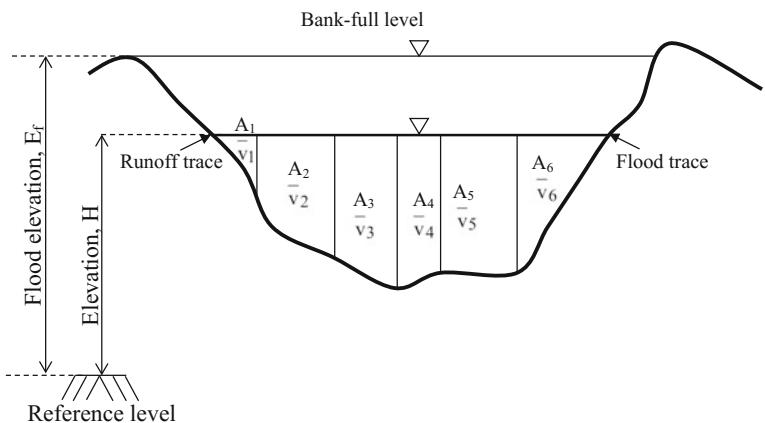
Cross-sections are useful means for different engineering standard calculations as bridge, dam, culvert, levee, pipe, tunnel, canal, structural design purposes, etc. For instance, flood inundation map preparation necessitates numerical information about cross-section shape, wet perimeter, and area under a given water level.

The cross-section slope is a part of the main channel slope around the cross-section location. In practical applications, the cross-section slope is calculated by taking into account a certain distance toward upstream and downstream directions, say about 100 m along the main channel thalweg as shown in Fig. 3.2.

Consideration of the notations in the figure leads to the cross-section slope,  $S_{cs}$ , by the following simple formulation.

$$S_{cs} = \frac{\Delta E}{2\Delta L} \quad (3.2)$$

Although the distance between the upstream and downstream measurement points is suggested as  $2 \times 100 = 200$  m, depending on the situation in the field works, one can take any convenient value instead of 100 m.

**Fig. 3.2** Cross-section slope**Fig. 3.3** Representative cross-section and sub-sections

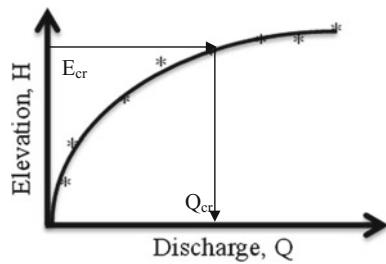
### 3.2.3 Rating Curve

This curve provides practical discharge calculation in a cross-section given the water elevation (or preferably depth). Figure 3.3 indicates a representative cross-section with different characteristics such as the bank-full level and its corresponding flood elevation with respect to a reference level.

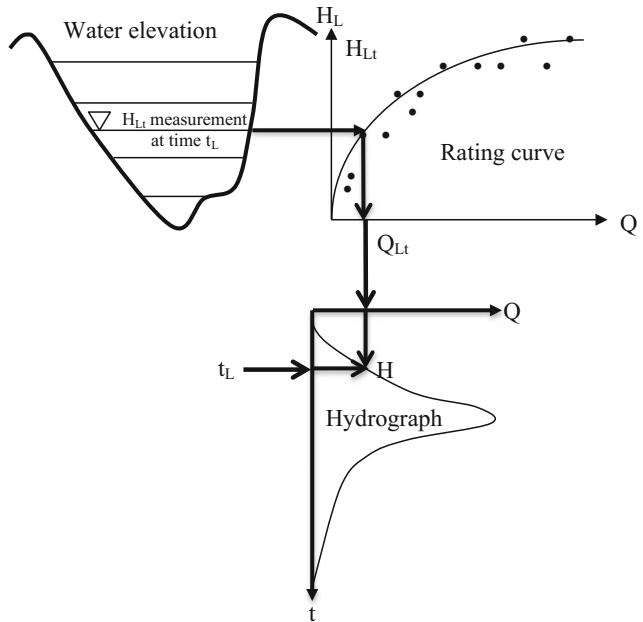
The sub-areas ( $A_1, A_2, \dots, A_6$ ) are in the form of a trapezium except on the left- and right-hand sides, which are in triangular forms. These basic geometrical shapes provide an easy way to calculate cross-section wet area as their summation. DEM provides basic data for cross-section shape, and the sub-areas are calculated through the following software at a set of water levels. Hence, a set of wet sub-section areas ( $A_{c1}, A_{c2}, \dots, A_{cn}$ ) becomes numerically available within the same cross-section at corresponding water-level elevation set ( $H_{c1}, H_{c2}, \dots, H_{cn}$ ).

The rating curve construction needs the discharge calculation through each wet sub-cross-section area corresponding to each water-level elevations. Since, by definition the discharge is the multiplication of area by velocity, it is necessary to calculate the velocity. For this purpose, the  $i$ -th cross-section sub-area average velocity,  $\bar{v}_i$ , is measured through a current meters, and hence, the average discharge is  $Q_i = A_i \bar{v}_i$  ( $i = 1, 2, \dots, 6$ ). The summation of these discharges gives the cross-section discharge,  $Q_{c1}$ . Repetition of the same procedure for a set of  $n$  water levels yields to  $n$  different cross-section discharges,  $Q_{c1}, Q_{c2}, \dots, Q_{cn}$ . The rating curve can be obtained by means of a smooth curve match through the scatter diagram of the cross-section water-level elevation versus discharge (Fig. 3.4).

In general, within the same drainage basin, each cross-section has a special rating curve. The rating curves are useful to estimate the discharge value even corresponding to the trace of flood on both sides of the cross-section. Provided



**Fig. 3.4** Representative rating curves



**Fig. 3.5** Rating curve–water elevation–hydrograph relationships

that the rating curve is available then from the water-level elevation measurement,  $H_{cr}$ , one can read discharge,  $Q_{cr}$ , on the horizontal axis.

In dry regions, natural channels most often do not have surface water, and therefore, the velocity calculations are achieved by suitable empirical formulations among which the Manning's equation is the most frequently adapted in practical applications (Sen 2017).

Figure 3.5 shows the practical construction of hydrograph variation in a cross-section related to available rating

curve through the water-level elevation measurements (Sen 2008).

To obtain a hydrograph first water-level measurement,  $H_{Lt}$ , is recorded at time  $t_L$ , as shown at the water elevation part of Fig. 3.5. The value of  $H_{Lt}$  is entered on the vertical axis of the rating curve to read the corresponding cross-section level discharge,  $Q_{Lt}$ , from the horizontal axis. Finally, two ( $Q_{Lt}$  and  $t_L$ ) values are entered into the hydrograph coordinate system to obtain point  $H$  representing the hydrograph. Repetition of the same procedure during the

raising and recessing cross-section water-level measurements helps to reach to the hydrograph generation.

According to what have been explained in this section, the cross-section calculations together with peak discharge estimation can be obtained by means the following two

software in MATLAB. The first software needs the cross-section coordinates and some of the cross-section features as input data. The second program has very detailed calculations and generates the cross-sections right from the DEM data.

```

function [TDDebi,TD,TZ]= RatingCurveandHydrograph(K,E,A,L,Lc)
% This program is written on 2 October 2014 by Zekai Sen
% K      : 2D cross-section abscissa and ordinate data matrix(x and y)
% E      : Cross-section slope
% A      : Cross-section area (m^2)
% n      : Number of water levels
% KEG    : Cross-section threshold level (m)
% DER    : Water depth (m)
% TD     : Peak discharge (Flood discharge) (m^3/s)
% TS     : Bank water level (flood bank level) (m)
% Savile: Rating curve depth (m)
M=0.0018; % Manning coefficient
x=K (:,1);
y=K (:,2);
% L=length(x);
scatter(x,y)
hold on
AR=0.01; %Spline space
xx=min(K(:,1)):AR:max(K(:,1)); %Spline horizontal axis data number
Lxx=length(xx);
yy=spline(x,y,xx); % Spline vertical axis data number
plot(xx,yy,'k')
hold on
[EKyy,EKI]=min(y); % The thalweg point in the cross-section and its position
[EBSol,ISol]=max(y(1:EKI)); % Left hand side maximum cross-section bank level
EBSolx=x(ISol);
[EBSag,ISag]=max(y(EKI:end)); % Right hand side maximum cross-section bank level
EBSagx=x(ISag);
EKkod=min(EBSol,EBSag); % The minimum elevation on the left and right hand side of the cross-section
Nsp=length(y);
k=0;
for i=1:EKI-1 % Loop for finding the threshold level point on the left hand side
    if y(EKI-i) <= y(EKI-i+1)
        k=k+1;
        YSo(k)=y(EKI-i);
        IYSo(k)=EKI-i;
    else
        end
    if k == 0
        YSol=y(1);
        EBy=y(1);
        EBx=x(1);
    else
        YSol=YSo(1);
        EBy=YSol;
        ISol=IYSo(1);
        EBx=x(ISol);
    end
end
..... Continued .....
```

```

l=0;
for i=EKI:Nsp-1 % Loop for finding the threshold level point on the right hand side
    if y(i+1) < y(i)
        l=l+1;
        YSa(l)=y(i);
        IYSa(l)=i;
    else
    end
    if l == 0
        YSag=y(end);
        ESy=y(end);
        ESx=x(end);
    else
        YSag=YSa(1);
        ESy=YSag;
        ISag=IYSa(1);
        ESx=x(ISag);
    end
end
if YSol <= YSag % Threshold level determination of the cross-section
ii=0;
for i=EKI:Nsp
    if y(i) > EBy
        ii=ii+1;
        ESII(ii)=i;
    else
    end
end
ESI=ESII(1);
ESy=y(ESI);
ESx=x(ESI);
else
    ii=0;
    for i=1:EKI-1
        if y(EKI-i) > ESy
            ii=ii+1;
            EBII(ii)=i;
        else
        end
    end
    EBI=EKI-EBII(1);
    EBy=y(EBI);
    EBx=x(EBI);
end
xE=EBx:1:ESx; % Threshold level x values
Sayi=length(xE);
EBS=min(EBy,ESy);
for i=1:Sayi
    yE(i)=EBS; % Threshold level y values
end
plot(xE,yE,'r-')
KEG=ESx-EBx; % Cross-section water level width

```

.....Continued.....

```

IB=0;
DER=EBS-EKyy; % The maximum water depth with respect to threshold level
Da=DER/100; % Depth increments
% n=3;
for k=1:100 %k=1:100*n
    Esik=EKyy+k*Da; % k/100;% Threshold level
    Seviye(k)=k*Da; % k/100
    for i=2:Lxx
        if yy(i-1)>= Esik && yy(i)<= Esik
            if IB == 0
                IB=i; % Treshold start
            end
        else
            if yy(i-1)<= Esik && yy(i)>= Esik
                IS=i-1; % Threshold end
            else
            end
        end
    end
    AL=0; % Area start
    IC=0; % Wet perimeter start
    for i=IB:IS
        if Esik-yy(i)>0 && Esik-yy(i-1)>0
            AL=AL+0.5*((Esik-yy(i))+(Esik-yy(i-1)))*AR; % Here, trapezium elements are used.
            IC=IC+sqrt((xx(i)-xx(i-1))^2+(yy(i)-yy(i-1))^2);
        end
    end
    HY=AL/IC; % Hydraulic radius
    Hiz=(1/M)*HY^0.66667*E^0.5; % Water speed
    D(k)=AL*Hiz; % Discharge calculation
end
MD=max(D);
TDebi=max(D); % Peak discharge
figure
[P] = CurveFitting(D,Seviye);
xlabel('Discharge (m^3/s)')
ylabel('Level (m)')
grid
%
% Discharge calculation according to Snyder method
%
SZ=0.75*1.5*(L*Lc/1000000)^0.3; % Snyder peak time
TZ=SZ/5.5; % Excess rainfall duration
TD=2.75*0.45*(A/1000000)/TZ; % Snyder peak discharge
Y=10*MD/(TD*A/1000000); %SD; (SD*A/1000000); % Rainfall=Threshold discharge/(SnyderDischarge*Area)
Hydrographs(TD,TZ,Y)
end

```

Supplementary to the previous software is the “CurveFitting” software, which is given in the following MATLAB program.

```

function [P] = CurveFitting(X,Y)
% This program is written by Zekai Sen on 23 November 2015
% It finds the best fitting mathematical expression between two sequences
% X : Horizontal axis variable
% Y : Vertical axis variable
% a,b,c,d : Parameters
% I = 1 LINEAR straight-line, y=a+bX
% I = 2 PARABOLIC line, y=a+bx+cx^2
% I = 3 CUBIC line, y=a+bx+cx^2+dx^3
% I = 4 QUADRATIC line, y=a+bx+cx^2+dx^3+ex^4
% I = 5 CEQUATIC line, y=a+bx+cx^2+dx^3+dx^4+ex^5
% I = 6 EXPONENTIAL line, y=aexp(bx)
% I = 7 LOGARITHMIC line, y=alog(x)+b
% I = 8 POWER line, y=ax^b
n=length(X);
scatter(X,Y, 'k')
hold on
% First order linear straight line
% I=1;
mX=mean(X);
mY=mean(Y);
mXY=mean(X.*Y);
mX2=mean(X.*X);
M=[1 mX; mX mX2];
PL=[mY mXY]/M;
YL=PL(1)+PL(2)*X;
EL=sum((Y-YL).^2);
% Second order parabola curve
% I=2;
mX3=mean((X.*X).*X);
mX4=mean((X.*X).*(X.*X));
mX2Y=mean(Y.*(X.*X));
M=[1 mX mX2;mX mX2 mX3;mX3 mX4];
PP=[mY mXY mX2Y]/M;
YP=PP(1)+PP(2)*X+PP(3)*X.^2;
EP=sum((Y-YP).^2);
% Third order cubic curve
% I=3;
mX5=mean((X.*X).*(X.*X).*X);
mX6=mean((X.*X).*(X.*X).*(X.*X));
mX3Y=mean(Y.*(X.*X).*X);
M=[1 mX mX2 mX3;mX mX2 mX3 mX4;mX2 mX3 mX4 mX5;mX3 mX4 mX5 mX6];
PC=[mY mXY mX2Y mX3Y]/M;
YC=PC(1)+PC(2)*X+PC(3)*X.^2+PC(4).*X.^3;
EC=sum((Y-YC).^2);
% Fourth order quadratic curve
% I=4;
mX7=mean((X.*X).*(X.*X).*(X.*X).*X);
mX8=mean((X.*X).*(X.*X).*(X.*X).*(X.*X));
mX4Y=mean(Y.*(X.*X).*X.*X);
M=[1 mX mX2 mX3 mX4;mX mX2 mX3 mX4 mX5;mX2 mX3 mX4 mX5 mX6;mX3 mX4 mX5 mX6 mX7;mX4 mX5 mX6 mX7 mX8];
PQ=[mY mXY mX2Y mX3Y mX4Y]/M;
YQ=PQ(1)+PQ(2)*X+PQ(3)*X.^2+PQ(4).*X.^3+PQ(5).*X.^4;
EQ=sum((Y-YQ).^2);
% Fifth order sequin curve
% I=5;
mX9=mean((X.*X).*(X.*X).*(X.*X).*(X.*X));
mX10=mean((X.*X).*(X.*X).*(X.*X).*(X.*X).*(X.*X));
mX5Y=mean(Y.*(X.*X).*X.*X);
M=[1 mX mX2 mX3 mX4 mX5;mX mX2 mX3 mX4 mX5 mX6;mX2 mX3 mX4 mX5 mX6 mX7;mX3 mX4 mX5 mX6 mX7 mX8;mX4 mX5 mX6 mX7 mX8;mX6 mX7 mX8 mX9;mX5 mX6 mX7 mX8 mX9 mX10];
PS=[mY mXY mX2Y mX3Y mX4Y mX5Y]/M;
YS=PS(1)+PS(2)*X+PS(3)*X.^2+PS(4).*X.^3+PS(5).*X.^4+PS(6).*X.^5;
ES=sum((Y-YS).^2);
% Exponential curve
% I=6;
mlogY=mean(log(Y));
mlogYX=mean(log(Y).*X);
M=[1 mX; mX mX2];
PE=[mlogY mlogYX]/M;
a=exp(PE(1));

```

.....Continued.....

```

YE=a*exp(PE(2)*X);
EE=sum((Y-YE).^2);
% Logarithmic curve
% I=7;
mlogX=mean(log(X));
mYlogX=mean(Y.*log(X));
mlogX2=mean(log(X).*log(X));
M=[1 mlogX;mlogX mlogX2];
PLo=[mY mYlogX]/M;
YLo=PLo(1)*log(X)+PLo(2);
ELo=sum((Y-YLo).^2);
% Power function (Double logarithmic)
% I=8;
mlogY=mean(log(Y));
mlogX=mean(log(X));
mlogYlogX=mean(log(X).*log(Y));
mlogX2=mean(log(X).*log(X));
M=[1 mlogX;mlogX mlogX2];
PK=[mlogY mlogYlogX]/M;
a=exp(PK(1));
YK=a*X.^^(PK(2));
EK=sum((Y-YK).^2);
[ME I]=min([EL EP EC EQ ES EE ELo EK]);
Hata=ME;
Xm=min(X);
YM=max(Y);
if I==1
    plot(X,YL,'r')
    P=PL;
    text(Xm+1,YM-3,['Y = ' num2str(PL(1)), '+', num2str(PL(2)), 'x']);
elseif I==2
    plot(X,YP,'r')
    P=PP;
    text(Xm+1,YM-3,['Y = ' num2str(PP(1)), '+', num2str(PP(2)), 'x', '+', num2str(PP(2)), 'x^2']);
elseif I==3
    plot(X,YC,'r')
    P=PC;
    text(Xm+1,YM-3,['Y = '
num2str(PC(1)), '+', num2str(PC(2)), 'x', '+', num2str(PC(2)), 'x^2', '+', num2str(PC(3)), 'x^3']);
elseif I==4
    plot(X,YQ,'r')
    P=PQ;
    text(Xm+1,YM-3,['Y = '
num2str(PQ(1)), '+', num2str(PQ(2)), 'x', '+', num2str(PQ(2)), 'x^2', '+', num2str(PQ(3)), 'x^3', '+', num2str(PQ(4)), 'x^4']);
elseif I==5
    plot(X,YS,'r')
    P=PS;
    text(Xm+1,YM-3,['Y = '
num2str(PS(1)), '+', num2str(PS(2)), 'x', '+', num2str(PS(2)), 'x^2', '+', num2str(PS(3)), 'x^3', '+', num2str(PS(4)), 'x^4', '+', num2str(PS(5)), 'x^5']);
elseif I==6
    plot(X,YE,'r')
    text(Xm+1,YM-3,['Y = ' num2str(a), 'exp(', num2str(PE(2)), ')x']);
elseif I==7
    plot(X,YLo,'r')
    P=PLo;
    text(Xm+1,YM-3,['Y = ' num2str(PLo(1)), 'log(', num2str(PLo(2)), ')x']);
elseif I==8
    plot(X,YK,'r')
    P=PK;
    text(Xm+1,YM-3,['Y = ' num2str(a), '^(', num2str(PK(2)), ')x', 'Power function']);
end
end

```

```

function [a,b,KEG,DER,SZ,SD,MD,YZ,Y]=CrossSectionBankDischargeRainfallIntensityFinder(K,E,M,A,L,Lc)
% This program is written by Zekai Şen on 2 October 2014.
% K : 2D vector for cross-section x and y coordinates
% E : Cross-section slope
% n : Water level number
% M : Manning coefficient
% AE : 2D vector for rating curve
% A : Drainage area(m2)
% L : Main channel length (m)
% Lc : The distance between the projection points of catchment centred on
%       the main channel and the outlet point(m)
% a and b : Cross-section rating curve parameters (Rating curve function
%           parameters)
% KEG : Cross-section bank width
% DER : Water depth in the cross-section (m)
% SZ : Peak discharge time (hour) - Snyder formulation
% SD : Peak discharge value (m3/sec) - Snyder formulation
% MD : The maximum discharge (bank full discharge) (m3/sec)
% Y : Excess rainfall duration(hour)
% Yagis : The amount of rainfall calculation in the software(Uniformly
%           distributed over the catchment area)
% Level=aD^b'dir. Herein, D is the discharge corresponding to water level.
%
%*****
% AKat : Runoff coefficient
% AYH : Excess rainfall volume(m3) is calculated on the assumption that the area
%       to the left (right) of peak discharge time is 37.5% (67.5%)
%       AYH=((SD*(2.66*SZ))*3600)/2
%
% AYY : Excess rainfall height(mm)          AYY=(AYH/Alan)*1000
% YM : Rainfall per square meter(mm)        YM=AYY/AKat
% YS : Rainfall intensity(mm/hour)          YS=YM/(2.66*SZ)
% SDS : Snyder discharge water level(m)-It is equivalent to water depth
%       that the Snyder discharge generates
% SDSKG: SnyderDischargeWaterLevelCross_sectionWidth (m)
%
%***** EXPLANATION *****
% This software can run whole data of any cross-section in one trial.
% All the corresponding data to the cross-section and the results are
% arranged in a single file. Names the graphs and then records.
% Discharge-level data are recorded in the form of Excel file separately.
%
% This software reads the data to the example file(such as Marmara_1_eninekesit.xls)
% All data set are arranged in this manner. Prior to program run adjust the
% MATLAB files (From current Directory) and run by pressing "Run" button.
% Subsequently for the whole drainage basin writes 12 for 12 cross sections.
% At the 55. line write the conclusions 'dizin' variable where all the
% results will be preserved.
% If there is any cross-section that needs to jump over without processing
% then at 64. line in for loop instead of havzano=1 enter the number of the
% next drainage basin and hence, you can rerun the software.
% *****

havzaadi=input('Enter the name of the main catchment(Example:Doğu Karadeniz) : ','s'); % This is
the root name of MATLAB file sequence.
havzasayisi=input('Enter sub-basin number (1-12): '); % Sequential catchment number in your
classor, one can enter any number between 1 and 12.

%***** Sequence where the results are stored *****
dizin='C:\Users\Zekai ŞEN\Desktop\zekai şen istanbul\Akarsu Havzaları Enine Kesit Verileri\Doğu
Akdeniz Havzası\'; % All results will be recorded here.
%*****



mkdir(dizin, [num2str(havzaadi) ' Havzası'])
mkdir(dizin, 'Result tables')

PATH1=[dizin num2str(havzaadi) ' Havzası'];
PATH2=[dizin 'Result tables\'];

..... continued .....
```

```

for havzano=1:havzasayisi    % Run the program up to the entered number of drainage basins.

dosya=xlsread([num2str(havzaadi) '_' num2str(havzano) '_eninekesit.xls'],1); % Control file names
and extensions and if necessary make corrections.
SutunIsimleri1=[{'No'} {'Ax'} {'Month'} {'Bx'} {'By'} {'Cross-Section Slope'} {'Area(m2)'} {'Main
Channel Length (m)'} {'Main Channel Slope'} {'Centroid Channel Length(m)'} {'Centroid Channel
Slope'}]; % These are the column names
SutunIsimleri2=[{'Rating Curve Parameter-a'} {'Rating Curve Parameter-b'} {'Bank-full width m)-
KEG'} {'Peak Discharge Time (hour)-SZ'} {'Peak Discharge (m3/sec)-SD'} {'Bank-full
Discharge(m3/sec)-MD'} {'Rainfall intensity(mm/h)'} {'Excess rainfall duration (hour)-YZ'}
{'Depth(m)-DER'} {'Snyder Discharge Water Level(m)-SDS'} {'Snyder Discharge Water Level (m)-
SDSKG'} {'Runoff Coefficient-AKat'} {'Excess Rainfall Volume m3)-AYH'} {'Excess rainfall Height
(mm)-AYY'} {'Rainfall Amount (mm)-YM'} {'Rainfall Intensity (mm/h)-YS'} ];
M=0,018;    %Manning coefficient
x=dosya(:,1);
y=dosya(:,2);
Ax=dosya(1,5);
Ay=dosya(1,6);
Bx=dosya(2,5);
By=dosya(2,6);
E=dosya(3,5);
A=dosya(4,5);
L=dosya(5,5);
Lc=dosya(6,5);
EL=dosya(8,5);
ELc=dosya(7,5);
AKat=dosya(9,5);
% In the following two lines physical values of the cross-sections are recorded into Excel file.
xlswrite([PATH1 num2str(havzaadi) ' Drainage_ ' results.xls'],SutunIsimleri1,'A1:K1');
xlswrite([PATH1 num2str(havzaadi) ' Drainage_ '
'sonuclar.xls'],[havzano,Ax,Ay,Bx,By,E,A,L,EL,Lc,ELc],[ 'A' num2str(havzano+1) ':' 'K'
num2str(havzano+1)]);

xlswrite([PATH2 num2str(havzaadi) ' Drainage_ ' results.xls'],SutunIsimleri1,'A1:K1');
xlswrite([PATH2 num2str(havzaadi) ' Drainage_ '
'sonuclar.xls'],[havzano,Ax,Ay,Bx,By,E,A,L,EL,Lc,ELc],[ 'A' num2str(havzano+1) ':' 'K'
num2str(havzano+1)]);

clear xx yy Lxx EKyy EKI EBSol ISol EBSolx EBSag ISag EBSagx EKkod xEkseni yEkseni Nsp EKder;
clear k YSo IYSol YSol EBy EBx ISol;
clear i YSa IYSa YSag ESY ESx ISag;
clear ESII ESI EBII EBI;
clear xE yE Sayi EBS KEG DER Da nDER n;
% Cross-section drawing
AR=0.01; %Spline basic interval
xx=min(x(:,1)):AR:max(x(:,1)); % Spline horizontal axis data
Lxx=length(xx); % Horizontal axis data number
yy=spline(x,y,xx); % Spline vertical axis data
h=figure;
plot(x,y,'k*',xx,yy); % Cross-section curve matched by Spline
hold on
title([num2str(havzaadi) ' Drainage' ' Cross-section sub-basin name:' num2str(havzano)]);
xlabel('Width(m)')
ylabel('Depth(m)')

[EKyy,EKI]=min(yy); % The value and position of the lowest point in the cross-section
[EBSol,ISol]=max(yy(1:EKI)); % The maximum cross-section elevation on the left hand side
EBSolx=xx(ISol);
[EBSag,ISag]=max(yy(EKI:end)); % The maximum cross-section elevation on the right hand side
ISag=ISag+EKI-1; % Herein "Isag" determines the position in the data sequence right from the
beginning.
EBSagx=xx(ISag);
EKkod=min(EBSol,EBSag); % The lowest threshold level on the right and left hand side of the cross-
section
EKder=EKkod-EKyy; % Bank-full level difference
% Cross-section boundary value line
xEkseni=min(x(:,1)):1:max(x(:,1));
yEkseni=EKkod;
plot(xEkseni,yEkseni,'g-','LineWidth',3);

```

..... continued .....

```

Nsp=length(yy);
k=0;
for i=1:EKI-1 % Finding loop for the left hand side of threshold elevation

    if yy(EKI-i)<= yy(EKI-i+1)
        k=k+1;
    YSo(k)=yy(EKI-i+1);% Value
    IYSo(k)=EKI-i+1;% Sequence number
    break
    else
    end
end
if k==0
    YSol=yy(1);% Value
    EBy=yy(1);% Value
    EBx=xx(1);% Value
else
    YSol=YSo(1);% Value
    EBy=YSol;% Value
    ISol=IYSo(1);% Sequence number
    EBx=xx(ISol);% Value
end
l=0;
for i=EKI:Nsp-1 % Finding loop for the right hand side of treswhold alavation
    if yy(i+1) <= yy(i)
        l=l+1;
    YSa(l)=yy(i); % Value
    IYSa(l)=i;      % Sequence number
    break
    else
    end
end
if l==0
    YSag=yy(end);
    ESy=yy(end);
    ESx=xx(end);
else
    YSag=YSa(1); % Value
    ESy=YSag;     % Value
    ISag=IYSa(1); % Sequence number
    ESx=xx(ISag); % Value
end

if YSol<=YSag    % Threshold level determination in the cross-section
ii=0;
for i=EKI:Nsp
    if yy(i)>=EBy
        ii=ii+1;
        ESII(ii)=i;
    break
    else
    end
end
ESI=ESII(1);% Sequence number
ESy=yy(ESI);% Value
ESx=xx(ESI);% Value
else
ii=0;
for i=1:EKI-1
    if yy(EKI-i)>ESy
        ii=ii+1;
        EBII(ii)=i;
    break
    else
    end
end

```

..... continued .....

```

EBI=EKI-EBII(1);% Sequence number
EBy=yy(EBI);% Value
EBx=xx(EBI);% Value
end

xE=EBx:1:ESx; % Threshold level x values
Sayi=length(xE);
EBS=min(EBy,ESy); % The lowest threshold value (Flow bed)
%for i=1:Sayi;
yE=EBS; %yE(i)=EBS; % Threshold level y values
%end
plot(xE,yE, 'r-','LineWidth', 4); % The smallest threshold line in the cross-section
saveas(h, [PATH1 num2str(havzaadi) ' Drainage_ ' num2str(havzano) '_Cross-section'], 'jpg');

KEG=ESx-EBx; % Cross-section water level width

IB=0;
DER=EBS-EKyy; % The maximum water depth at the bank level of the cross-section
Da=DER/100; % Depth increments
nDER=DER;
if DER<0.2 && EKder>=2
    nDER=2;
end
n=min(nDER, 3);
D=0;
Seviye=0;
for k=1:100*n
    Esik=EKyy+k/100;% Bank level
    Seviye(k)=k/100;
    for i=2:Lxx
        if yy(i-1)>= Esik && yy(i)<= Esik
            if IB == 0
                IB=i; % Bank-full start
            end
        else
            if yy(i-1)<= Esik && yy(i)>= Esik
                IS=i-1; % Bank end
            else
            end
        end
    end
    AL=0; % Area start
    IC=0; % Wet perimeter start
    for i=IB:IS
        if Esik-yy(i)>0 && Esik-yy(i-1)>0
            AL=AL+0.5*((Esik-yy(i))+(Esik-yy(i-1)))*AR; % Here, trapezium elements are used.
            IC=IC+sqrt((xx(i)-xx(i-1))^2+(yy(i)-yy(i-1))^2);
        end
    end
    HY=AL/IC; % Hydraulic radius
    Hiz=(1/M)*HY^0.66667*E^0.5; % Hydraulic speed
    D(k)=AL*Hiz; % Discharge calculation
    Debi_Seviye(k,1)=[D(k)]; % Two lines are for discharge and level calculations.
    Debi_Seviye(k,2)=[Seviye(k)]; % For transfer to Excel table.
end
figure
b1=mean(log(Seviye).*log(D))-mean(log(D))*mean(log(Seviye));
b2=mean(log(D).^2)-mean(log(D))^2;
b=b1/b2;
a=exp(mean(log(Seviye))-b*mean(log(D)));
MD=max(D); % Bank-full discharge
xEDebisi=0:0.01:MD;
yEDebisi=a*xEDebisi.^b;
h=figure;
plot(xEDebisi,yEDebisi,'r')
hold on
scatter(D,Seviye,'.k')
xlabel('Discharge (m^3/sec)')
ylabel('Level (m)')
grid

```

..... continued .....

```

title([num2str(havzaadi) ' Drainage' ' Discharge Level Graph Sub-basin no:' num2str(havzano)]);
saveas(h,[PATH1 num2str(havzaadi) ' Drainage_' num2str(havzano) '_Discharge level'],'jpg')
hold off
%
% Snyder method is used in discharge calculations
%
SZ=0.75*1.5*(L*Lc/1000000)^0.3; % Snyder peak discharge time
YZ=SZ/5.5; % Excess rainfall duration
SD=2.75*0.45*(A/1000000)/SZ; % Snyder peak discharge
Y=10*MD/SD; % (SD*A/1000000); % Rainfall=Bank discharge/(SnyderDischarge*Area)
AYH=((SD*(2.66*SZ))*3600)/2; % Excess rainfall volume(m³)
AYY=(AYH/A)*1000; % Excess rainfall height mm
YM=AYY/AKat; % Rainfall amount per square meter (mm)
YS=YM/(2.66*SZ); % Rainfall intensity(mm/hour)

% In this part Snyder discharge cross-section water level is calculated.
for j=1:100*n
    if D(j)<=SD
        SDS=j;
    else
        break
    end
end
SDS=SDS/100; %Snyder discharge water level(m)

% In this part Snyder discharge, water level and cross-section width are calculated.
x=dosya(:,1);
y=dosya(:,2);
AR=0.01; %Spline basic interval
xx=min(x(:,1)):AR:max(x(:,1)); % Spline horizontal axis data
yy=spline(x,y,xx); % Spline vertical axis data
[EKyy,EKI]=min(yy);
Nsp=length(yy);
for t=EKI:Nsp
if yy(t)<=SDS+EKyy
SDKGI sag=t;
else
    break
end
end

SDKGsag=xx(SDKGI sag);
for r=1:EKl
if yy(r)>=SDS+EKyy
SDKGI sol=r;
else
    break
end
end

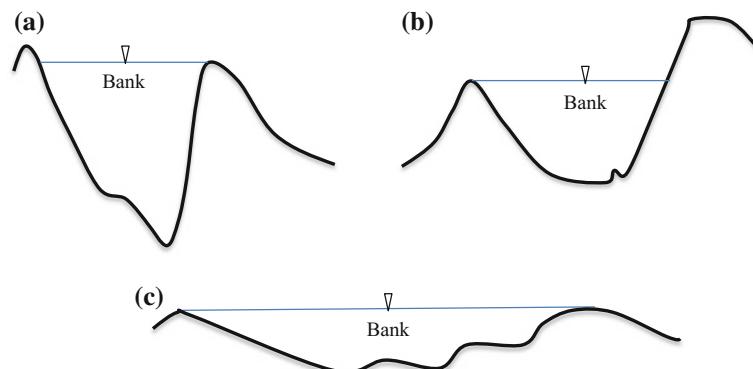
SDKGsol=xx(SDKGI sol);

SDSKG=SDKGsag-SDKGsol; %Snyder discharge water level cross-section width (m).

xlswrite([PATH1 num2str(havzaadi) ' Drainage_ ' results.xls'],SutunIsimleri2,'L1:AA1');
xlswrite([PATH1 num2str(havzaadi) ' Drainage_ '
'sonuclar.xls'],[a,b,KEG,SZ,SD,MD,Y,YZ,DER,SDS,SDSKG,AKat,AYH,AYY,YM,YS],[ 'L' num2str(havzano+1) ':' 
'AA' num2str(havzano+1)]);
xlswrite([PATH1 num2str(havzaadi) ' Drainage_' num2str(havzano) '_Discharge Level.xls'],Debi_Seviye);

xlswrite([PATH2 num2str(havzaadi) ' Drainage_ ' results.xls'],SutunIsimleri2,'L1:AA1');
xlswrite([PATH2 num2str(havzaadi) ' Drainage_ '
'results.xls'],[a,b,KEG,SZ,SD,MD,Y,YZ,DER,SDS,SDSKG,AKat,AYH,AYY,YM,YS],[ 'L' num2str(havzano+1) ':' 
'AA' num2str(havzano+1)]);
end
close all
end

```



**Fig. 3.6** Bank-full cross-section and flood: **a** right bank, **b** left bank, **c** both banks

In the following more detailed software is presented about the same problem for cross-section bank-full discharge calculation after rainfall intensity determination.

### 3.3 Floods

A flood is a type of runoff that transgresses the bank-full level of a cross-section so that the extra water spreads on either sides of the cross-section as in Fig. 3.6. This definition indicates that the flood is not only the consequence of peak discharge, but the discharge that is more than the capacity of a cross-section along the water movement pathway.

In these bank-full cross-sections, the flood may occur on the left, right, or both sides. In practical applications, the question is where to take the cross-section? This depends on the project purpose. For flood control purposes, it must be taken in the upper elevation zones of drainage basin to protect human activity and settlement areas (villages, towns, and cities), agricultural lands, military quarters, and the like. Hence, one can calculate or better observe the water table

rise through cameras and data loggers instantaneously and see the water-level closeness to the bank-full level. This provides two types of information.

- (1) The closeness of water level to the bank-full level provides information for early warning flood danger appreciation,
- (2) At the bank-full level flood starts to take place.

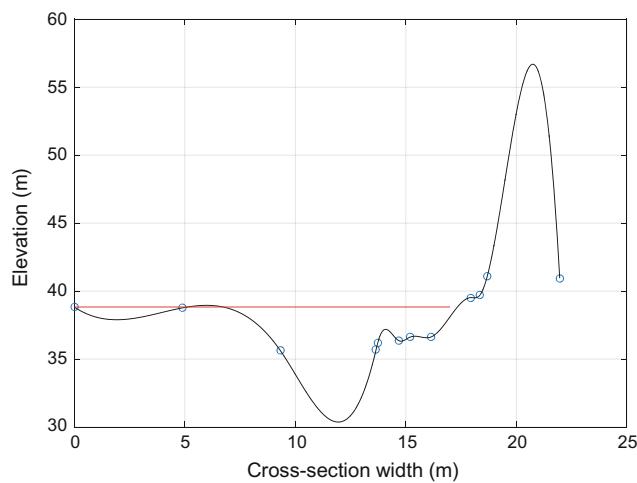
Cross-section numerical information can be obtained either from the available DEM data in the office or by lidar sensors in a finer manner from field measurements. The following software helps to depict the cross-section bank-full level according to the empirical Manning's formulation, which can be found in any hydrology textbook (Hann et al. 1982; Wanielista et al. 1997; Şen 2008). In Fig. 3.7, a cross-section is given as the software result with the bank-full water level.

The following software written in MATLAB language helps to calculate discharge at a set of water levels.

```

function [D,Seviye]=RatingCurve(K,E,m,M)
% This program is written by Zekai Sen on 18 May 2014.
% K : 2D cross-section abscissa and ordinate data matrix(x and y)
% E : Cross-section slope
% m : Depth reduction number
% M : Manning coefficient
% AE : 2d vector that includes the rating curve
x=min(K(:,1)):1:max(K(:,1));
y=K(:,2);
plot(x,y)
AR=0.01; %Spline intervals
xx=min(K(1,:)):AR:max(K(1,:)); %Spline horizontal axis data
Lxx=length(xx);
yy=spline(x,y,xx); % Spline vertical axis data
plot(x,y,'k*',xx,yy); % Spline curve match shape
[EKyy,I]=min(yy); % Thalweg point elevation and position
[EBSol,ISol]=max(yy(1:I)); % Left hand side bank elevation
[EBSag,ISag]=max(yy(I:end)); % Right hand side bank elevation
Esik=min(EBSol,EBSag); % Threshold level
Derinlik=Esik-EKyy; % Flood depth
SeviyeDusmesi=Derinlik/m; % Here m indicates water level fall amount
% Quantities at bank-full level
ES=Esik-yy;
ES=ES(ES>0);
L=length(ES);
AL=0; % Area start
IC=0; % Wet perimeter start
for i=2:L
    AL=AL+0.5*(yy(i)-yy(i-1))*AR;
    IC=IC+sqrt((xx(i)-xx(i-1))^2+(yy(i)-yy(i-1))^2);
end
Seviye(1)=Esik;
Alan(1)= AL;
IslakCevre(1)=IC;
HY(1)=Alan(1)/IslakCevre(1);
Hiz(1)=(1/M)*HY(1)^0.66667*E^0.5; % Manning speed
D(1)=Alan(1)*Hiz(1); % Discharge calculation
for j=2:m
    Seviye(j)= Seviye(j-1)-SeviyeDusmesi;
    yys=Seviye(j)-yy
    for i=1:Lxx
        if yys(i) > 0
            N(i)=i;
        else
            end
    end
    NI=N(N>0)
    In=min(NI)
    yys=yys(yys>0);
    LL=length(yys);
    AL=0; % Area start
    IC=0; % Wet perimeter start
    for i=2:LL
        AL=AL+0.5*(yys(i)-yys(i-1))*AR;
        IC=IC+sqrt((xx(i)-xx(i-1))^2+(yys(i)-yys(i-1))^2);
    end
    Alan(j)=AL;
    IslakCevre(j)=IC;
    HY(j)=Alan(j)/IslakCevre(j);
    Hiz(j)=(1/M)*HY(j)^0.66667*E^0.5; % Manning speed
    D(j)=Alan(j)*Hiz(j); % Discharge calculation
end
end

```



**Fig. 3.7** Bank-full water level for a cross-section

Flood hydrographs can be calculated practically from the unit hydrograph (UH) concept, which is a very useful method. The UH is defined as the hydrograph that is resultant from the uniform and homogeneously distributed 1 cm depth of rainfall layer over the whole drainage basin. In general, each drainage basin depending on its geological and geomorphological features has separate UHs. However, engineers have reduced it even down to the simplest form as the dimensionless unit hydrograph (DUH) by dividing the discharge values in a hydrograph by the peak discharge and the time duration by peak discharge time. Furthermore, many DUHs in a drainage basin or region appear as very similar to each other, and therefore, engineers took the arithmetic average of the ordinates and reached to a standard DUH ([Sen 2017](#)).

Practical elegance of DUH concept is its ability to help to calculate the peak discharge and its occurrence time, and then by multiplying the DUH abscissa by peak discharge time and ordinates by peak discharge, one can obtain the corresponding hydrograph. In hydrology literature, there are different empirical formulations for peak discharge and its time calculations among which are the Snyder, Soil Conservation Service (SCS), and similar approaches that can be found in any textbook on hydrology (Maidment and Mays 1988; Linsley et al. 1982; Muthreja 1986; Subramanp 1994).

In the following, three MATLAB programming language written programs are given for hydrograph calculation starting from DUH values.

```

function UnitHydrographandHydrographs(TZD,YS)
% This program is written by Zekâi Şen on 19 October 2014 at 21:23 O'clock
% B      : Unit hydrograph (UH) coordinate discharges
% B1     : Dimensionless UH discharges
% B2     : Dimensionless UH times
% TD    : Peak discharge (m3/sec)
% TZ    : Peak discharge time (hour)
% YS    : Rainfall intensities are YS=[1 2 3 4 5 6].
B2=[0 0.2349 0.4273 0.5829 0.7068 0.8768 0.9303 1.0000 0.9921 0.9329 0.8423 0.7393 0.6358 0.6023
0.5382 0.4230 0.3269 0.2494 0.2327 0.1631 0.1129 0.0774 0.0526 0.0356 0.0239 0.0106 0.0047 0.0008
0.0000];
B1=[0.0 0.1 0.2 0.3 0.4 0.6 0.7 1.1 1.2 1.5 1.8 2.1 2.4 2.5 2.7 3.1 3.5 3.9 4.0 4.5 5.0 5.5 6.0
6.5 7.0 8.0 9.0 11.0 14.0];
B=[B1;B2];
ns=length(YS); % Intensity data number
nk=length(TZD); % Cross-section number
for i=1:nk
    figure
    HZ=B(1,:)*TZD(i,1); % Adjusts the necessary time axis.
    HD=B(2,:)*TZD(i,2); % Adjusts the necessary discharge axis.
    for j=1:ns
        YHD=HD*YS(j); % Hydrograph adjustment according to rainfall intensity
        plot(HZ,YHD)
        hold on
    end
    xlabel('Time (hour)')
    ylabel('Discharge (m3/sn)')
    title('CROSS-SECTION HYDROGRAPHS')
    grid
    figure
end
end

```

```

function Hydrographs(TD,TZ,YS)
% This program is written by Zekâi Şen on 19 October 2014 at 21:23 o'clock
% B      : Unit hydrograph (UH)ordinate discharges
% B1     : Dimensionless UH discharges
% B2     : Dimensionless UH times
% TD    : Peak discharge (m3/sec)
% TZ    : Peak time (hour)
% YS    : Rainfall intensities are YS=[1 2 3 4 5 6]
B2=[0 0.2349 0.4273 0.5829 0.7068 0.8768 0.9303 1.0000 0.9921 0.9329 0.8423 0.7393 0.6358 0.6023
0.5382 0.4230 0.3269 0.2494 0.2327 0.1631 0.1129 0.0774 0.0526 0.0356 0.0239 0.0106 0.0047 0.0008
0.0000];
B1=[0.0 0.1 0.2 0.3 0.4 0.6 0.7 1.1 1.2 1.5 1.8 2.1 2.4 2.5 2.7 3.1 3.5 3.9 4.0 4.5 5.0 5.5 6.0 6.5
7.0 8.0 9.0 11.0 14.0];
B=[B1;B2];
ns=length(YS); % Intensity data number
nk=length(TD); % Cross-section number
% figure
for i=1:nk
    HZ=B(1,:)*TZ(i); % Adjusts the necessary time axis
    HD=B(2,:)*TD(i); % Adjusts the necessary discharge axis
    for j=1:ns
        YHD=HD*YS(j); % Hydrographs adjusted according to rainfall discharges
        plot(HZ,YHD)
        hold on
    end
    xlabel('Time (hour)')
    ylabel('Discharge (m3/sec)')
    title('HYDROGRAPHS')
    grid
end
end

```

@Seismicisolation

```
% for u=1:(SZ(i)+6/2)*5+1
% Hydrograph_elements(u,1)=[w(u)];
% Hydrograph_elements(u,2)=[ttt(u)];
% Hydrograph_elements(u,3)=[qqp(u)];
% Hydrograph_elements(u,4)=[Qs(u)];
% end
format
h=figure;
%ARTIS=0.01; % Spline interval
%TT=min(w(1,:)):INCREMENT:max(w(1,:)); % Spline horizontal axis data
%QQ=spline(w,Qs,TT); % Spline vertical axis data
plot(w,Qs,'b-','LineWidth',2);
xlabel('Zaman (sa)')
ylabel('Debi (m^3/s)')
grid
title([char(havzaadi(1,k)) ' Drainage basin' ' 6 hour unit hydrograph Sub-basin No:'
num2str(i)]);
saveas(h,[PATH1 char(havzaadi(1,k)) ' Drainage_ num2str(i) _Unit hydrograph'], 'jpg')

%xlswrite([PATH1 char(havzaadi(1,k)) ' Havzası_ num2str(i) 'hidrograf
Değerleri.xls'],SutunIsimleri,'A1:C1');
%xlswrite([PATH1 char(havzaadi(1,k)) ' Havzası_ num2str(i) 'hidrograf
Değerleri.xls'],hidrografin_elemanlari);
end
close all
end
```

### 3.3.1 Flood Risk

Extreme events such as floods should be managed by risk consideration, because they are record-breaking phenomena in nature, and hence, in scientific calculations always there are errors. Although the risk level is defined in Chap. 2 based on the precipitation records through a theoretical PDF, in hydrology the risk definition is rather different for any given cross-section. The closer is the water level to the bank-full level the more is the risk for flood and inundation. Hence, the risk level,  $R_L$ , based on the water level in a cross-section can be scrutinized by comparison of the actual water level,  $L_A$ , with the bank-full level,  $L_B$ , through the relative error definition as follows.

$$R_L = 100 \frac{L_B - L_W}{L_B} \quad (3.3)$$

This yields a value between 0 and 100. At this point depending on the expert view of surface water flow at the cross-section, one can classify the risk levels into three or five classes. In the following cross-section representatively three water-level risks are shown (Fig. 3.8).

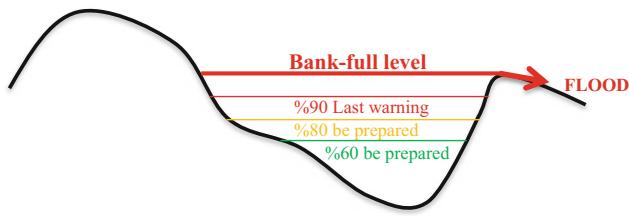
The levels in this figure help to guide administrators, operators, technicians, decision makers, and engineers to watch possible flood early warning developments depending

on the risk levels, and hence, one is able to inform the people prior to flood and inundation incidences.

After the cross-section overtopping for flood occurrence, the extra water that goes on either sides of the cross-section covers lower elevation areas (inundation area) with water, and hence, some land pieces on both sides remain under water, and accordingly, the necessary precautions should be taken for the next flood occurrence. Among the precautions are the levees, channel widening, and water diversions to convenient areas, bends, levees, and small dams for flood protections, and in arid regions these dams are useful for groundwater recharge.

The following MATLAB software gives flood risk information depending on the relative error (Eq. 3.3) closeness of the actual water level to the bank-full water level prior to flood occurrence.

In any cross-section, for water-level or discharge calculations one needs to calculate the water-level and discharge relationship, which is referred to as the “rating curve” explained in Sect. 3.2.3. The following software calculates the rating curve depending on the cross-section data and provides its functional form with the corresponding figure. The following “function [WarningType] = FloodEarly Warning(K,E,A,L,Lc,YS)” program calls the MATLAB program from Sect. 3.2.3 “function [TDebi,TD,TZ] = RatingCurveandHydrograph(K,E,A,L,Lc)”.



**Fig. 3.8** Cross-section risk levels

```

function [WarningType] = FloodEarlyWarning(K,E,A,L,Lc,YS)
% This program is written by Zekai Sen on 24 December 2015 at 10:04 O'clock
% Program warns for flood given rainfall intensity
% K : Cross-section X and Y values
% A : Drainage basin area(m^2)
% E : Cross-section slope
% TS : Bank-full level (m)
% TmaxD : Bank-full discharge (m^3/sec)
% EKS : Minimum level (m)
% EKD : Minimum discharge (m^3/sec)
% YS : Rainfall intensity (mm/hour)
[TDebi,TD,TZ]= RatingCurveandHydrograph(K,E,A,L,Lc) % This program is available in section 3.2.3
SUS1=0.6*TmaxD; % Light yellow warning level
SUS2=0.7*TmaxD; % Dark yellow warning level
SUS3=0.8*TmaxD; % Light red warning level
SUS4=0.9*TmaxD; % Dark yellow warning level
SUS5=0.95*TmaxD; % Red warning level
% Rational method discharge calculation
AD=0.35*A*(YS/1000) % Flow discharge (m^3/hour)
if AD < SUS1
    WarningType='%60 DANGER'
elseif AD> SUS1 && AD < SUS2
    WarningType='%70 DANGER'
elseif AD> SUS2 && AD < SUS3
    WarningType='%80 DANGER - VISIT THE LOCATION'
elseif AD> SUS3 && AD < SUS4
    WarningType='%90 DANGER - WAIT AT THE LOCATION AND WARN LOCAL PEOPLE'
else
    WarningType='%95 DANGER ALARM'
end
end
end

```

### 3.3.2 Probable Maximum Flood (PMF)

After the completion of the PMP calculations as in Chap. 2, Sect. 2.5, it is possible to convert the PMP values through a convenient rainfall–runoff relationship leading to the peak discharge calculations and thereof to corresponding hydrographs. For this purpose, the following steps are necessary.

- (1) The annual daily maximum rainfall (ADMR) implies that the rainfall duration is 24 h, which may not be the case in many arid and semiarid regions, where the thunderstorms continue for 3–5 h at the maximum. It is, therefore, necessary to calculate the duration of the rainfall within a day. This can be achieved empirically according to the Snyder (1938) method, which gives the time to peak discharge,  $T_p$ , by taking into consideration

the longest drainage channel length,  $L$ , and the length,  $L_c$ , of this channel from the projection of the centroid point on the channel to the outlet point,

- (2) Although the peak discharge,  $Q_p$ , can be calculated according to various methodologies (Sen 2008), herein, the simplest one, the rational method is considered, which can be used with confidence especially in small drainage areas,
- (3) It is necessary to obtain the distribution of the flood with time and a convenient methodology for the calculation of the resultant hydrograph. For this purpose, dimensional unit hydrograph (DUH) approach is adopted.

In order to achieve all these calculations, the software in MATLAB language is presented below.

```

function [HT,HD] = ProbableMaximumFloodGraph(tp,qp)
% This program plots the hydrograph for a given catchment peak discharge
% and time to peak discharge values
% Written by Zekai Sen on 29th March 2014
% tp is time to peak
% qp is the peak discharge
% DUHD : Dimensionless UH discharges
% DUHT : Dimensionless UH times
% HD : Hydrograph discharge values (m3/sec)
% HT : Hydrograph time values(hour)
% In the following are dimensionless unit hydrograph times and discharges
DUHD=[0 0.2349 0.4273 0.5829 0.7068 0.8768 0.9303 1.0000 0.9921 0.9329 0.8423 0.7393
0.6358 0.6023 0.5382 0.4230 0.3269 0.2494 0.2327 0.1631 0.1129 0.0774 0.0526 0.0356
0.0239 0.0106 0.0047 0.0008 0.0000];
DUHT=[0.0 0.1 0.2 0.3 0.4 0.6 0.7 1.1 1.2 1.5 1.8 2.1 2.4 2.5 2.7 3.1 3.5 3.9 4.0 4.5
5.0 5.5 6.0 6.5 7.0 8.0 9.0 11.0 14.0];
% B=[B1;B2];
HT=tp*DUHT; % Hydrograph time base series
HD=qp*DUHD; % Hydrograph discharge series
plot(HT,HD, 'k','LineWidth',2);
xlabel('Time (hour)')
ylabel('Discharge (m3/sec)')
title('Probable maximum flood (PMF) hydrograph')
box on
grid on
end

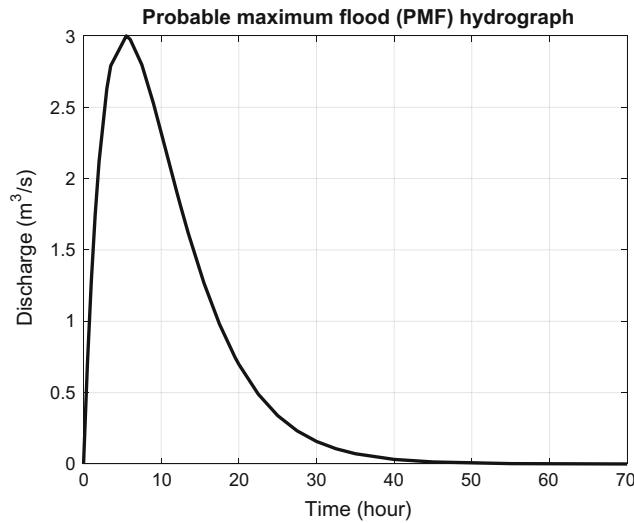
```

As an example, Fig. 3.9 presents the output from this MATLAB program flood hydrograph for  $t_p = 3$  h and  $q_p = 5 \text{ m}^3/\text{s}$ .

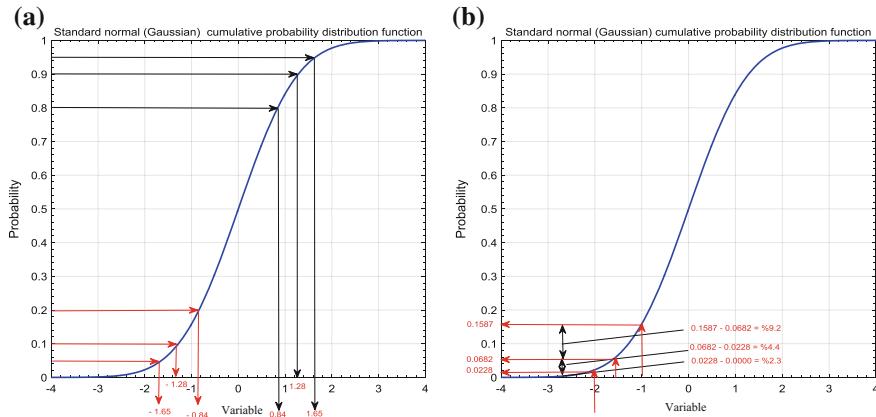
### 3.4 Risk Calculation

It is the amount of uncertainty that can be accepted in any human-made structural design; after all, these structures will resist natural events such as precipitation and its subsequent

form of surface water (runoff) occurrences, which cannot be predicted certainly in temporal and spatial scales. In hydrology, all natural events have completely uncertain parts that cannot be accounted deterministically by any scientific methodology. For instance, even after the estimation of the peak discharge, it is not guaranteed that in the future, it will not be overtapped. Whatever the refinement of the scientific methodology, there is always an unexplainable part, and therefore, the designer, planner, or engineer has to take a certain risk level in his/her calculations. The risk magnitude



**Fig. 3.9** Probable maximum flood hydrograph



**Fig. 3.10** Risk levels: **a** from risk levels to variables, **b** from variable values to risk levels

can be calculated from the available data, if any, through a theoretical PDF. In the hydrological calculations the most recommended risk level is at the maximum 10%, but in some more sensitive works it may be taken as 5%. In Fig. 3.10, a common methodology of risk attachment from, say, normal (Gaussian) cumulative PDF is shown for representation purpose.

At this point, it is useful to relate the risk,  $R$ , to the design period,  $T$  (recurrence interval), which are inversely related to each other by definition as,

$$R = \frac{1}{T} \quad (3.4)$$

Herein,  $R$  is defined as the probability of exceedence once during the whole design period,  $T$ . The design intensity,  $x_d$ ,

can be related to a convenient PDF,  $f(x)$  according to the following integration.

$$R = \int_{x_d}^{+\infty} f(x) dx \quad (3.5)$$

The PDF of any hydro-meteorological event helps to calculate the magnitude of an event for a given set of risk levels (Chap. 2). For this purpose, the following MATLAB software provides risk calculation based on six different CDFs (Gamma CDF, Weibull, log-normal, normal (Gaussian), exponential and extreme value).

```

function [Risk] = RiskCalculations(V,G,Xtitle)
% This program is written on 21 April 2016 by Zekai Sen
% V : Data vector
% G : Probability distribution function indicator
%     G = 1 Gamma CDF
%     G = 2 Weibull CDF
%     G = 3 Logarithmic CDF
%     G = 4 Normal CDF
%     G = 5 Exponential CDF
%     G = 6 Extreme value (Gumbel) CDF
n=length(V); % Data number
figure
mV=min(V);
MV=max(V);
d=(MV-mV)/100;
x=0.9*mV:d:1.1*MV;
p=(1:1:n)/(n+1);
scatter(sort(V),1-p,'k*');
hold on
grid on
box on
xlabel(Xtitle)
ylabel('Exceedence probability')
% Theoretical probability distribution function selection START
if G==1
    title('Gamma probability distribution function')
    pr=gamfit(V);
    y=gamcdf(x,pr(1),pr(2));
    plot(x,1-y,'r','LineWidth',2)
    legend('Data','Gamma PDF')
    text(1.02*mV, 0.90,['Location parameter = ' num2str(pr(1))])
    text(1.02*mV, 0.85,['Shape parameter = ' num2str(pr(2))])
    Risk=gaminv([0.99 0.95 0.90 0.85],pr(1),pr(2));
elseif G==2
    title('Weibull probability distribution function')
    pr=wblfit(V);
    y=wblcdf(x,pr(1),pr(2));
    plot(x,1-y,'r')
    text(1.1*mV, 0.9,['Location parameter = ' num2str(pr(1))])
    text(1.1*mV, 0.8,['Shape parameter = ' num2str(pr(2))])
    Risk=wblinv([0.99 0.95 0.90 0.85],pr(1),pr(2));
elseif G==3
    title('Log-normal probability distribution function')
    pr=lognfit(V);
    y=logncdf(x,pr(1),pr(2));
    plot(x,1-y,'r')
    text(1.1*mV, 0.9,['Location parameter = ' num2str(pr(1))])
    text(1.1*mV, 0.8,['Shape parameter = ' num2str(pr(2))])
    Risk=logninv([0.99 0.95 0.90 0.85],pr(1),pr(2));
elseif G==4
    title('Normal (Gauss) probability distribution function')
    MO=mean(V);
    SO=std(V);
    y=normcdf(x,MO,SO);
    plot(x,1-y,'r')
    text(1.1*mV, 0.9,['Location parameter = ' num2str(pr(1))])
    text(1.1*mV, 0.8,['Shape parameter = ' num2str(pr(2))])
    Risk=norminv([0.99 0.95 0.90 0.85],pr(1),pr(2));
elseif G==5

```

.....Continued.....

```

title('Exponential probability distribution function')
pr=expfit(V);
y=expcdf(x,pr);
plot(x,1-y,'r')
text(1.1*mV, 0.9,['Location parameter = ' num2str(pr(1))])
text(1.1*mV, 0.8,['Shape parameter = ' num2str(pr(2))])
Risk=expinv([0.99 0.95 0.90 0.85],pr(1),pr(2));
else
    title('Extreme value probability distribution function')
    pr=evfit(V);
    y=evcdf(x,pr(1),pr(2));
    plot(x,1-y,'r')
    text(1.1*mV, 0.9,['Location parameter = ' num2str(pr(1))])
    text(1.1*mV, 0.8,['Shape parameter = ' num2str(pr(2))])
    Risk=evinv([0.99 0.95 0.90 0.85],pr(1),pr(2));
end
end

```

On the other hand, risk-based regression methodology application possibilities can be achieved by the software in MATLAB presented below.

In the first software “function [Risk] = RiskCalculations (V,G,Xtitle)” the successive event occurrences are assumed as independent from each other. However, most often

```

function [Risk] = RiskBasedRegression(X,N)
% This program is written by Zek-i Şen on 16 December 2012 Sunday
% The program calculates risk levels and plots them on a Cartesian
% coordinate system for prediction purposes. Given a certain column all
% other columns are plotted as scatter diagram on the Cartesizn coordinate
% system.
% X is the data of n columns and m rows
% N is the selection column
% Return periods
R=[2 5 10 25 50 75 100 150 200 250 400 500 600 750 1000];
% Column and row sizes
ns=size(X);
n=ns(1,1); % Row number
m=ns(1,2); % Column number
% Select the column that will appear on the horizontal axis as input
XSelection=X(:,N);
% If you want to plot the change of independent variables onto dependent
% variable then activate the following 5 statements
% figure
% for i=1:m
%     scatter(XSelection,X(:,i));
%     hold on
% end
for i=1:n
    par=gamfit(X(i,:));
    for j=1:15
        Risk(i,j)=gaminv(1-1/R(j),par(1),par(2));
    end
end
figure
for j=3:3
    scatter(XSelection,Risk(:,j),'.');
    hold on
end
end

```

hydrological time series has serial dependence, for which the following MATLAB software is available.

```

function [R T RR]=SimpleRiskCalculationDependentSeries(p,r,n)
% This program calculates the number of crossings in independent processes.
% Sen, Z. (1999). Simple risk calculation in dependent hydrologic series.
% Journal of Hydrological Sciences, Vol. 44, No. 6,
% 871-878.
% p = Probability of surplus
% r = Autorun coefficient
% n = sample length
% Bivariate conditional probabilities
q=1-p;
PSGS=r;
PSGD=(p/q)*(1-r);
PDGD=1-PSGD;
PDGS=1-r;
% Risk period
n1=n-1;
R=1-q*(PDGD)^n1;
% Return period
T=q*q/((PDGD)*p*PDGS);
end

```

### 3.5 Droughts

All over the world the effect of drought is in steady increase; it is not yet well understood, because commonly agreed drought definition is not available and its impacts are not assessed yet satisfactorily. Definitions are based on basic careers (hydrological, agricultural, meteorological, geographical) or according to industrial, energy generation, water supply, navigation and recreational regions. Roughly droughts are defined as the temporary reduction in the rainfall, runoff, and soil moisture amounts, which are related to the climatology of the region. Especially, dry climates are prone to more pronounced drought effects due to the soil moisture deficiency and high variability in the rainfall occurrence frequency and amounts.

Nonexistence of precise and universally accepted drought definition is another problem that adds to the confusion about universally accepted drought classification and severity. Since drought is climate dependent, its definition should also consider the local and regional climatic features (Chaps. 2 and 7). On the other hand, drought implies different meanings as implications to water managers, engineers, agriculturalists, hydroelectric power plant operators, medical specialists, physiologists, and wildlife biologists.

There are various drought definitions in the literature. Generally, it is defined as an extended period of rainfall deficit during which water resources and agricultural

harvests are severely curtailed. Wilhite and Glantz (1985) have classified drought definitions as conceptual, i.e., relatively vague (fuzzy) and operational manners, which is

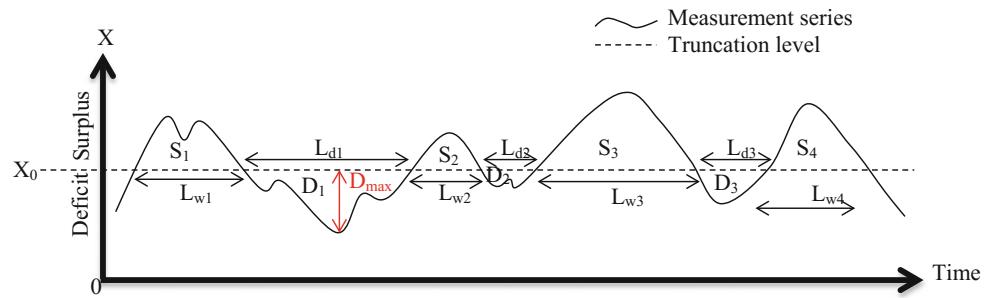
meant to provide specific guidance on aspects of drought onset, severity, and termination. Frequently asked questions for objective drought description are as follows.

- (i) When is the starting time of drought?
- (ii) How severe is the drought?
- (iii) When is the termination time of the drought?

Droughts have a “creeping” feature, which is very gradual, and hence, their developments are slow and have prolonged existence, sometimes over many years and even decades. Droughts are not confined with local topographic features or geological structures, but they are more extensive over large areas. In practice, it is most often very difficult and almost impossible to pinpoint drought beginning and, hence, their distinctions from human-induced desertification (Sen 2015).

#### 3.5.1 Wet and Dry Periods

In general, the term “wet spells” corresponds to the periods of excessive precipitation amounts that cause water surpluses and occasional floods or flash floods. On the other hand, “dry spells” are the periods of precipitation deficit resulting in occasional water shortages, droughts, arid conditions, and desertification. The fundamentals concerning wet and dry spells and droughts are presented in detail by Sen (2015). In general, wet and dry spells are identified by

**Fig. 3.11** Wet and dry periods

duration, intensity, severity, and spatio-temporal extent. Monthly precipitation records provide basic data for detecting first meteorological and subsequently hydrological and agricultural droughts, which are defined as the inability to provide society with sufficient water supply (Choi et al. 2013; Mishra and Singh 2010; Şen 2015).

One of the characteristics of many precipitation regimes throughout the world is that wet spells occur in a variety of durations. Monthly precipitation records may be analyzed simply using a certain risk level (Sect. 3.5), which helps to identify also the critical magnitudes of monthly wet and dry spells at a truncation level. Full information about the wet and dry spell numbers become more pressing in water resources management projects, agricultural planning, flood and drought studies. The geographical distribution of these meteorological parameters helps to plan and manage water resources related projects. Figure 3.11 gives detailed information about wet and dry periods of a time series,  $X_i$  ( $i = 1, 2, \dots, n$ ) at a truncation (threshold) level,  $X_0$ .

In this figure, sequences of surplus and deficit periods are given on a certain truncation level,  $X_0$ , which may be the arithmetic average or any other supply or demand level. Above and below given truncation levels, there are various dry and wet period specifications.

In Fig. 3.11 dry periods are referred to as the drought representatives and their lengths ( $L_{d1}, L_{d2}, \dots$ ) are the drought durations. In the figure, three distinctive drought durations are shown. These durations are very important in cases of water scarcity, and in a given record, there are many of them. The question is which one of these drought durations to adapt in practical applications. The sequence of such durations provides a basis for probabilistic and statistics

methodological applications. One may take the arithmetic average as the representative drought duration, but in practical applications, it is better to adapt a value that corresponds to a certain risk level such as 5 or 10%. All what has been said for drought duration the same arguments are valid for wet durations.

In Fig. 3.11, each deficit spell has differences of  $X$  values from  $X_0$  and their summation is referred to as the drought magnitude,  $D_M$ , which can be expressed as,

$$D_M = \sum_{i=1}^{L_d} (X - X_0) \quad (3.6)$$

It shows the total deficit amount along any drought duration. Among all the drought magnitudes within the historical data, the maximum one may be taken as the most dangerous case that may appear in the future. There is another description of the drought, which corresponds to the maximum deficit,  $D_{\max}$ , value during any drought duration.

After knowing the drought magnitude and its duration, the drought intensity,  $D_{II}$ , is defined as the drought magnitude divided by the corresponding drought duration. For instance, according to Fig. 3.11 the first drought intensity,  $D_{II1}$ , is calculated as,

$$D_{II1} = \frac{D_{\max}}{L_{d1}} \quad (3.7)$$

It is possible to calculate each one of the drought quantities at risk levels (0.50, 0.20, 0.10, 0.04, 0.02, and 0.01) as in Chap. 2 corresponding to a set of return periods (2-year, 5-year, 10-year, 25-year, 50-year, and 100-year).

### 3.5.2 Drought Indices

There are many drought indices in practical use for different purposes including engineering, agricultural, irrigation, meteorology, soil moisture, and other activities. The most widely used alternatives are either the classical approach as explained above or the standard precipitation index, which is effective in meteorological applications for depicting the effects of meteorological droughts (Chap. 2).

It is possible to define objectively the drought properties by considering the features in Fig. 3.11 (Şen 2015).

- (i) A “wet” state ( $X_i > X_0$ ) occurs at any time interval,  $i$ , with its “surplus” amount as  $(X_i - X_0) > 0$ ,
- (ii) Otherwise a “dry” state ( $X_i < X_0$ ) exists with its corresponding “deficit” amount  $(X_0 - X_i) < 0$ ,
- (iii) A “wet spell” is defined as the delimitation of consecutive uninterrupted wet spells by at least one “wet” state. If the dry state locations at the beginning and the end of a “wet spell” are  $(X_i < X_0)$  and  $(X_j < X_0)$ , respectively, then the duration of this wet spell is defined quantitatively as  $(j - i)$ ,
- (iv) Likewise, “dry spell” is defined as an uninterrupted sequence of “dry” states that are delimited at the two ends by at least one “wet” state. If the “wet” state locations at the beginning and the end of a “dry spell” are “wet” states as  $(X_k > X_0)$  and  $(X_l > X_0)$ , then the “dry spell” duration is equal to  $(l - k)$ ,
- (v) If there is a transition from a “dry spell” to subsequent “wet spell,” then such a transition point is defined by two successive states as “dry” ( $X_i < X_0$ ) and “wet” ( $X_{i+1} > X_0$ ). This means that at  $i$ -th location there is an up-crossing point,  $u_i$ ,
- (vi) Similarly, any transition between “wet spell” and the following “dry spell” implies a down-crossing,  $d_i$ , point defined by a “wet” state ( $X_i > X_0$ ) and the next “dry” state ( $X_{i+1} < X_0$ ),
- (vii) Among the “dry spell” durations at a given truncation level there is one, which is the maximum (minimum), and this duration is referred to as the “critical dry duration” (“critical wet duration”). Such critical periods play a significant role in any capacity design in water structures (dams, canals, dikes, levees, etc.) and especially in the water resources operations and management,
- (viii) The summation of surpluses (deficits) along any wet duration (dry duration) is referred to as the “surplus

magnitude,”  $S_M$  (“drought magnitude,”  $D_M$ , as in Eq. 3.6). These are also known as the total surplus and total deficit amounts, respectively, in practical studies. It is also possible to consider the “gross deficit,”  $D_G$  (“gross surplus,”  $S_G$ ) by considering the totals over the whole record period. Comparison of these two gross values leads to the following three alternatives,

- (a) If  $D_G = S_G$  then at this location, temporarily, the summation of deficits is in balance with surpluses during the whole record or planned period considered. It is possible to balance the whole deficit occurrence during this duration by surplus amounts provided that the surpluses are stored in convenient storage. For instance, if the water demand level of a settlement area is considered as the truncation level then the total surplus around this level can meet the whole deficit (water shortage) occurrences without any water shortage provided that proper storage units are constructed. Such a situation corresponds to ideal reservoir design, which is referred to as the Ripple diagram in hydrology (Chow 1964; Linsley 1967),
- (b) If  $D_G > S_G$ , then at this location one expects drought effect, and in order to offset such a case, it is necessary to transfer water from other locations,
- (c) If  $D_G < S_G$ , there is a wet spell expectation at this location with extra water amounts and it may be possible to transfer the extra water to nearby water-shortage or drought-stricken areas.

Each one of the drought definitions can be calculated, if there is a reliable sequence of data at hand with objectively defined threshold level. It is also possible to make future predictions of these quantities for different return periods (design durations) such as 5-year, 10-year, 25-year, 50-year, or 100-year (Şen 2015). These are already studied analytically by Şen (1976, 1977) through mathematical statistical, probability, and stochastic methodologies.

Two pieces of software below in MATLAB are for the drought feature calculations. The first one makes the necessary drought duration, magnitude, maximum deficit, and intensity calculations, and the next is called from this program for determination of risk-level drought features.

```

function [C,S,SL,SM,SI,D,DL,DM,DI]=
InnovativeDroughtAnalysis(T,X,Title,Xtitle,Ytitle,Unit1,Unit2,Unit3)
% This program has been written by Zekai Sen in 1978 in Fortran language
% and converted to MATLAB in 2002
% BURADA VERİLEN BİRİMLERE DİKKAT ET
% It is about drought analysis of a given time series
% IMPORTANT NOTICE: LET THE SEQUENCE START WITH A VALUE MORE THAN THE
% AVERAGE
% X is a given time series
% T is time interval series (years, months, etc.)
% X0 is the truncation level
% C is the crossing vector
% S (D) is the surplus (deficit) sum vector
% SL (DL) is the surplus (deficit) length vector
% SM (DM) is the surplus (deficit) magnitude vector
% SI (DI) is the surplus (deficit) intensity vector
% Unit1 is the basic for instance mm or m^3/s
% Unit2 is the duration for instance month or year
% Unit3 is the intensity for instance mm/month, mm/year, (m^3/s)/month,
% (m^3/s)/year
Xo=mean(X);
figure
plot(T,X,'k','LineWidth',2)
title(Title)
xlabel(Xtitle)
ylabel(Ytitle)
grid on
box on
SF={['Crossing number','Surplus amounts','Surplus length','Surplus magnitude','Surplus intensity']}; %
Surplus names
DF={['Deficit amounts','Deficit length','Deficit magnitude','Deficit intensity']}; % Deficit names
n=length(X);
j=0;
for i=2:n
    sign =(X(i-1)-Xo)*(X(i)-Xo);
    if sign < 0
        j=j+1;
        C(j)=i-1;
    else
    end
end
j1=j-1;
if X(1) >= Xo % Initial stage is wet
    l=1;
else
    l=2;
end
if l == 1
    for i=1:C(1)
        surplus(i)=(X(i)-Xo);
    end
    S(1)=sum(surplus(1:C(1)));
    SM(1)=max(surplus(1:C(1)));
    SL(1)=C(1);
    SI(1)=S(1)/SL(1);
    m=1;
    for i=2:2:j1
        m=m+1;
        for k=(C(i)+1):C(i+1)
            surplus(k)=(X(k)-Xo);
        end
        S(m)=sum(surplus(C(i)+1:C(i+1)));
        SM(m)=max(surplus(C(i)+1:C(i+1)));
        SL(m)=C(i+1)-C(i);
        SI(m)=S(m)/SL(m);
    end
end
.....Continued.....

```

```

m=0;
for i=1:2:j1
    m=m+1;
    for k=(C(i)+1):C(i+1)
        deficit(k)=(X(k)-Xo);
    end
    D(m)=abs(sum(deficit(C(i)+1:C(i+1))));
    DM(m)=abs(min(deficit(C(i)+1:C(i+1))));
    DL(m)=C(i+1)-C(i);
    DI(m)=D(m)/DL(m);
end
else
for i=1:C(1)
    deficit(i)=(Xo-X(i));
end
D(1)=abs(sum(deficit(1:C(1))));
DM(1)=abs(min(deficit(1:C(1))));
DL(1)=C(1);
DI(1)=D(1)/DL(1);
m=1 ;
for i=2:2:j1
    m=m+1;
    for k=(C(i)+1):C(i+1)
        deficit(k)=(Xo-X(k));
    end
    D(m)=abs(sum(deficit(C(i)+1:C(i+1))));
    DM(m)=abs(min(deficit(C(i)+1:C(i+1))));
    DL(m)=C(i+1)-C(i);
    DI(m)=D(m)/DL(m);
end
for i=(C(end)+1):n
    deficit(i)=(Xo-X(i));
end
D(m+1)=abs(sum(deficit((C(end)+1):n)));
DM(m+1)=abs(min(deficit((C(end)+1):n)));
DL(m+1)=n-C(end)+1;
DI(m+1)=D(m)/DL(m);
m=0;
for i=1:2:j1
    m=m+1;
    for k=(C(i)+1):C(i+1)
        surplus(k)=(X(k)-Xo);
    end
    S(m)=sum(surplus(C(i)+1:C(i+1)));
    SM(m)=max(surplus(C(i)+1:C(i+1)));
    SL(m)=C(i+1)-C(i);
    SI(m)=S(m)/SL(m);
end
end

```

.....Continued.....

```

if X(end) >= Xo
    for i=(C(end)+1):n
        surplus(i)= (X(i)-Xo);
    end
    S(m+1)=sum(surplus(C(end)+1:n));
    SM(m+1)=max(surplus(C(end)+1:n));
    SL(m+1)=n-C(end);
    SI(m+1)=S(m+1)/SL(m+1);
else
    for i=(C(end)+1):n
        deficit(i)= (X(i)-Xo);
    end
    D(m+1)=abs(sum(deficit(C(end)+1:n)));
    DM(m+1)=abs(min(deficit(C(end)+1:n)));
    DL(m+1)=n-C(end);
    DI(m+1)=abs(D(m+1)/DL(m+1));
end
[V,I] = ProbabilityDistributionFunctionChoice(C',SF(1),Title,' ');
[V,I] = ProbabilityDistributionFunctionChoice(S',SF(2),Title,Unit1);
[V,I] = ProbabilityDistributionFunctionChoice(SL',SF(3),Title,Unit2);
[V,I] = ProbabilityDistributionFunctionChoice(SM',SF(4),Title,Unit1);
[V,I] = ProbabilityDistributionFunctionChoice(SI',SF(5),Title,Unit3);
[V,I] = ProbabilityDistributionFunctionChoice(abs(D'),DF(1),Title,Unit1);
[V,I] = ProbabilityDistributionFunctionChoice(abs(DL'),DF(2),Title,Unit2);
[V,I] = ProbabilityDistributionFunctionChoice(abs(DM'),DF(3),Title,Unit1);
[V,I] = ProbabilityDistributionFunctionChoice(abs(DI'),DF(4),Title,Unit3);
end

```

```

function [V,I] = ProbabilityDistributionFunctionChoice(D,Xtitle,StName,Unit)
% This program is written on 13 September 2015 Sunday by Zekai Sen from
% Istanbul Technical University e-mail:zsen@itu.edu.tr
% There are Gamma, Log-Normal, Extreme Value (EV Gumbel)and Generalized
% Extreme Value (GEV, Pearson) probability distribution functions % This program produces Intensity-
frequency curve for any given time
% duration
% D : Time series data
% Xtitle : Time series data variable name with unit
% R : Risk levels
% StName : Station name
% V : It is the least sum of squares of probability deviations
%       from the theoretical probability distribution
% I : The number of PDF
%       If I = 1 Gamma PDF
%       If I = 2 Log-Normal PDF
%       If I = 3 Extreme value (Gumbel)PDF
%       If I = 4 Generalized extreme value (Pearson III)PDF
%       If I = 5 Weibull PDF
Risk=0.001:0.001:0.999;
R=[1-Risk(500) 1-Risk(200) 1-Risk(100) 1-Risk(40) 1-Risk(20) 1-Risk(10) 1-Risk(4) 1-Risk(2)];
n=length(D);
DM=1.1*max(D);
Dm=min(D);
x=Dm:0.1:DM;
pp=(1:1:n)/(n+1); % Data probability in ascending order
p=1-pp'; % Data probability in descending order
SD=sort(D); % Sorted time series in ascending order
pgam=gamfit(D); % Gamma PDF parameters
ygam=1-gamcdf(x,pgam(1),pgam(2));
ptgam=1-gamcdf(SD,pgam(1),pgam(2));
ppt2gam=(p-ptgam).^2;
GTest=mean(ppt2gam);
plon=lognfit(D); % Log-Normal PDF parameters
ylon=1-logncdf(x,plon(1),plon(2));
.....Continued .....
```

```

ptlon=1-logncdf(SD,plon(1),plon(2));
ppt2lon=(p-ptlon).^2;
LNTTest=mean(ppt2lon);
pevd=evfit(D); % Extreme value PDF parameters
yevd=1-ecdf(x,pevd(1),pevd(2));
ptevd=1-ecdf(SD,pevd(1),pevd(2));
ppt2evd=(p-ptevd).^2;
EVTTest=mean(ppt2evd);
pgev=gevfit(D); % Generalized extreme value PDF parameters
ygev=1-gevcdf(x,pgev(1),pgev(2),pgev(3));
ptgев=1-gevcdf(SD,pgev(1),pgev(2),pgev(3));
ppt2gев=(p-ptgев).^2;
GEVTest=mean(ppt2gев);
pwbl=wblfit(D); % Weibull PDF parameters
ywbl=1-wblcdf(x,pwbl(1),pwbl(2));
ptwbl=1-wblcdf(SD,pwbl(1),pwbl(2));
ppt2wbl=(p-ptwbl).^2;
WBLTest=mean(ppt2wbl);
rgam=gaminv(R,pgam(1),pgam(2));
rlon=logninv(R,plon(1),plon(2));
revd=evinv(R,pevd(1),pevd(2));
rgev=gevinv(R,pgev(1),pgev(2),pgev(3));
rwbl=wblinv(R,pwbl(1),pwbl(2));
[V I]=min([GTest LNTTest EVTTest GEVTest WBLTest]);
if I == 1
yf=ygam;
rf=rgam;
pr=pgam;
PR='Gamma PDF';
elseif I ==2
yf=ylon;
rf=rlon;
pr=plon;
PR='Log-normal PDF';
elseif I == 3
yf=yevd;
rf=revd;
pr=pevd;
PR='Gumbel';
elseif I == 4
yf=ygev;
rf=rgev;
pr=pgev;
PR='Pearson PD';
else
yf=ywbl;
rf=rwbl;
pr=pwbl;
PR='Weibull PDF';
end
figure()
scatter(SD,p,'k*')
title(StName)
xlabel(Xtitle)
ylabel('Exceedence probability')
hold on
grid on
box on

```

.....Continued.....

```

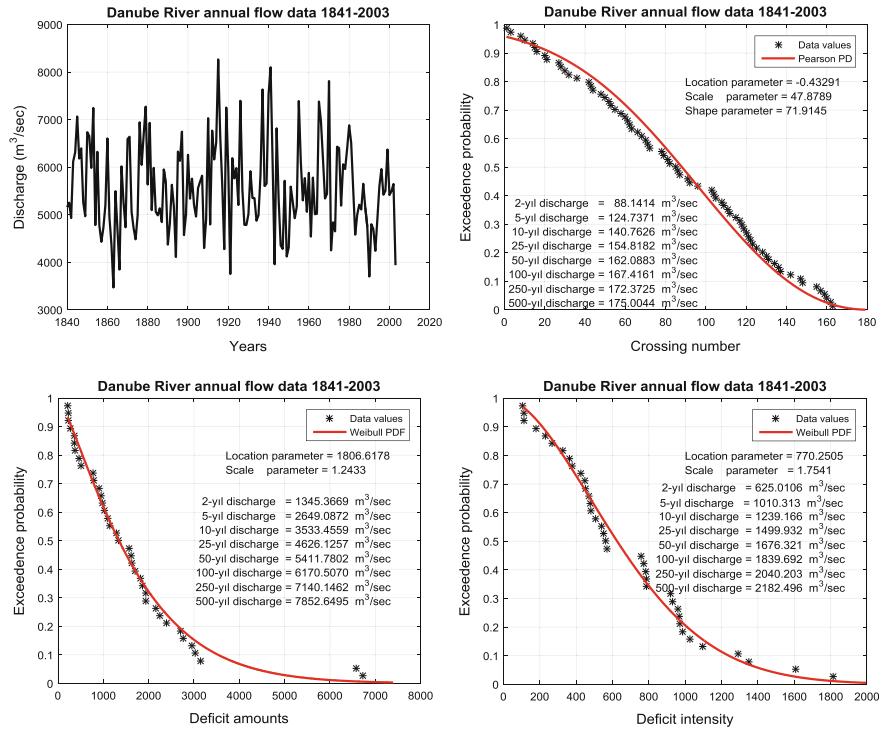
plot(x,yf,'LineWidth',2,'Color','r') % Theoretical PDF plot
legend('Data values',PR,'Location','Northeast')
text(0.5,0.8,[Location parameter = ' num2str(pr(1),'%0.1f')], 'Units','normalized')
text(0.5,0.75,[Scale parameter = ' num2str(pr(2),'%0.1f')], 'Units','normalized')
text(0.60,0.60,[' 0.50-risk = ' num2str(rf(1),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.55,[' 0.20-risk = ' num2str(rf(2),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.50,[' 0.10-risk = ' num2str(rf(3),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.45,[' 0.04-risk = ' num2str(rf(4),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.40,[' 0.02-risk = ' num2str(rf(5),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.35,[' 0.01-risk = ' num2str(rf(6),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.30,[' 0.004-risk = ' num2str(rf(7),'%0.1f'), ' ',Unit], 'Units','normalized')
text(0.60,0.25,[' 0.002-risk = ' num2str(rf(8),'%0.1f'), ' ',Unit], 'Units','normalized')
if I == 4
    text(0.5,0.70,[Shape parameter = ' num2str(pr(3),'%0.1f')], 'Units','normalized')
else
end
end
%end

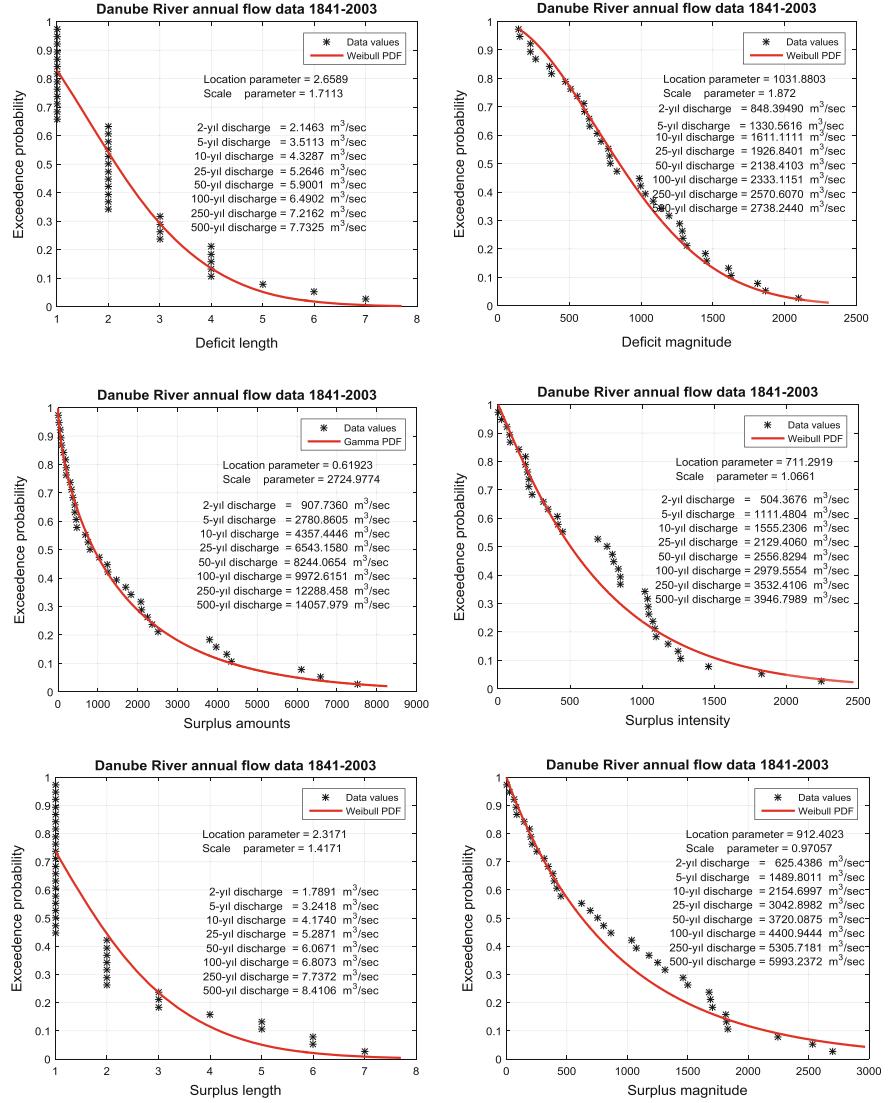
```

The application of the previous two MATLAB programs on the monthly river discharges of the Danube River results in the following graphs automatically (Fig. 3.12).

On the other hand, the storage volume calculation based on a certain threshold level can be achieved by the following MATLAB program.

**Fig. 3.12** Various drought quantities for Danube River



**Fig. 3.12** (continued)

```

function [T,V] = StorageCalculation(X)
% This program is written on 28 September 2014.
% X : Time series for storage calculation
% T : Truncation level - Demand
% V : Storage volume
%
%
n=length(X);
Ort=mean(X); % Arithmetic average
Ssa=std(X); % Standard deviation
s=12; % Level sampling number
XM=max(X);
Xm=min(X);
DA=(XM-Xm)/12; % Variation interval
for i=1:10 % Calculations are achieved only at 10 levels with exception of XM and Xm
    T(i)=Ort+i*DA;
    XK=cumsun(X-T(i));
    XKM=max(XK);
    XKm=min(XK);
    V(i)=XKM-XKm;
end

```

Entrance of Danube River monthly flows into this program yields the following ten truncation levels and storage volumes, respectively.

Mishra, A. K., and Singh, V. P. (2010). A review of drought concepts. *J. Hydrol.* 391, 202–216. <https://doi.org/10.1016/j.jhydrol.2010.07.012>. Maidment, D.R., Mays, L.W., (1988). *Applied Hydrology*, McGraw-Hill Book Company, New York.

---

$$T = [5.9665 \quad 6.3662 \quad 6.7658 \quad 7.1655 \quad 7.5652 \quad 7.9648 \quad 8.3645 \quad 8.7642 \quad 9.1638 \quad 9.5635] \times 10^3$$

and

$$S = [0.6627 \quad 1.2988 \quad 1.9503 \quad 2.6017 \quad 3.2532 \quad 3.9047 \quad 4.5561 \quad 5.2076 \quad 5.8590 \quad 6.5105] \times 10^5$$


---

## References

- Campbell, K. W. (2002). Engineering models of strong ground motion, in *Earthquake Engineering Handbook*, W. F. Chen and C. Scawthorn (Editors), CRC Press, Boca Raton, Florida, 5-1–5-76.
- Choi SJ, Shin IJ, Je KH, Min EK, Kim EJ, Kim HS, Choe S, Kim DE, Lee DK (2013) Hypoxia antagonizes glucose deprivation on interleukin 6 expression in an Akt dependent, but HIF-1/2α independent manner. *PLoS One* 8(3):e58662. doi: <https://doi.org/10.1371/journal.pone.0058662>.
- Chow, V.T. 1964. *Handbook Of Applied Hydrology*. McGraw- Hill, New York.
- Hann, C.T., Johnson, H.P., Brakensiek, D.L., (1982). Hydrologic modelling of small watersheds. *ASAE Monograph No. 5*. St. Joseph, MI.
- Linsley, R.K., 1967. The relation between rainfall and runoff: review paper. *Journal of Hydrology*, 5, pp. 297–311.
- Linsley, R.K., Kohler, M.A., Paulhus, J.L.H., (1982). *Hydrology for Engineers*, McGraw-Hill Book Company, New York.
- Muthreja, K.N., (1986). *Applied Hydrology*, Tata McGraw Hill Publishing Company Limited, New Delhi.
- Snyder, F.F., (1938). Synthetic unit graphs. *Transactions, American Geophysics Union*, Vol. 19: 447–454.
- Subramany, K., (1994). *Engineering Hydrology*, Tata McGraw Hill Publishing Company Limited, New Delhi.
- Sen, Z. (1976) Wet and dry periods of annual flow series. *J. Hydraul. Engng Div. ASCE* 102(HY10), 1503–1514.
- Sen, Z. (1977) Run sums of annual flow series. *J. Hydrol.* 35, 311–325.
- Sen, Z., (2008). *Wadi Hydrology*. Taylor and Francis Group, CRC Press, Boca Raton: 247 pp.
- Sen, Z., (2015). *Applied Drought Modeling, Prediction, and Mitigation*. Elsevier Publishing Co., 412 pp.
- Sen, Z., (2017). *Flood Modeling, Prediction, and Mitigation*. Springer Verlag, 422 pp.
- Wanielista, M.P., Kersten, R., and Eaglin, R., (1997). *Hydrology: Water Quantity and Quality Control*. Wiley, New York.
- Wilhite, D.A. and M.H. Glantz. 1985. Understanding the Drought Phenomenon: The Role of Definitions. *Water International*, 10(3), 111–120.



# Hydrogeology

# 4

## 4.1 General

Hydrogeology is a branch of earth sciences that deals quantitatively and qualitatively with groundwater flow through porous, fractured, and karstic (solution cavity) geological formations. The flow mechanism is the same as for any other fluid such as natural gas and petroleum with only differences in the density. Hydrogeology is the science that lies in the intersection of hydrology and geology. It is concerned more with groundwater flow evaluation, assessment, and numerical evaluation to identify the water-related parameters of the geological formations, which are categorized into two broad parts as the pervious and impervious layers. Pervious geological layers are divided into saturated and unsaturated layers. The saturation layers are the main quantitative hydrogeology topics, because their water-related parameters are of great concern in groundwater resources planning, operation, and management studies. Terminologically, pervious geological layers are referred to as “aquifers,” which may have porous, fractured, and karstic geological types. In practical groundwater resources evaluations, there are two major hydrogeological parameters for aquifer groundwater potentiality assessment, which are the transmissivity, storage and leakage coefficients. For groundwater movement apart from the interconnected voids within the geological layers, there must be groundwater-level difference between any two points and the slope of the groundwater surface causes to movement, and therefore, in order to standardize this groundwater-level difference, it is divided by the distance between the two points, which is then referred to as the hydraulic gradient. Logically, the more the hydraulic gradient the faster is the groundwater flow.

There are different aquifer parameter identification methodologies depending on the type of aquifer, whether confined, unconfined or leaky; infinitesimally small or large diameter wells; main or observation wells; whole duration or late time aquifer (pumping) test data. Detailed information

about the aquifer and well type calculation methodology are given in textbooks about the aquifer parameters (Freeze and Cherry 1979; Fetter 2001; Sen 1995, 2015).

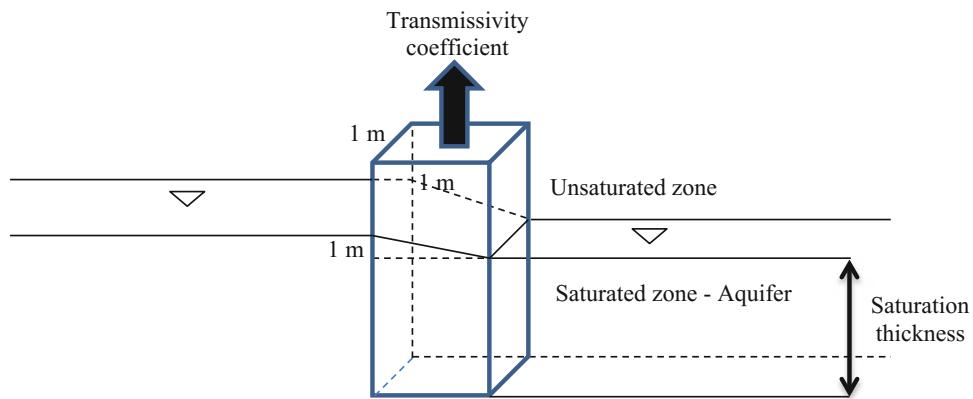
This chapter provides a set of simple MATLAB language programs for each of these alternatives so that the aquifer parameters, namely storage, transmissivity, leakage coefficients, and others, can be calculated automatically provided that time-drawdown measurements are available.

## 4.2 Aquifer Parameters

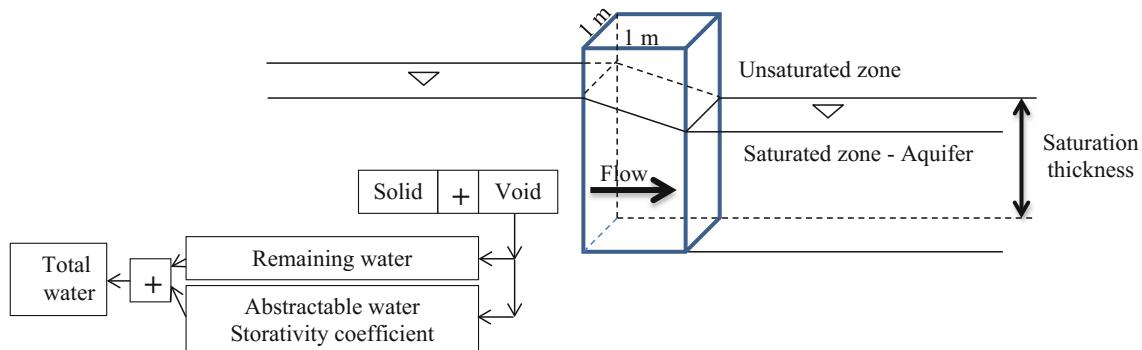
By definition, the transmissivity coefficient is the amount of water that can be taken from (or given into) an aquifer under unit hydraulic gradient from the whole saturation thickness and unit horizontal area of the saturation surface (Fig. 4.1).

The storativity coefficient is a part of the void volume within the porous, fractured or karstic aquifers, and generally, it represents the amount of abstractable water. This implies that all the voids in the saturation zone are full of water, but in practice, it is not possible to abstract all the water, because there is some part that is in contact with the voids’ surface as a result of adhesion forces. These explanations indicate that the storage coefficient is a part of the well-known porosity, and hence, it assumes theoretical values between zero and one. However, practically it can never be close to 1 because of the adhesion forces. Experiences by many researchers and hydrogeologists indicate that it may have values between 0.01 and 0.45. The storage coefficient is the amount of abstractable water from the saturation zone from unit horizontal area and the whole saturation thickness (Fig. 4.2).

As it will be explained later in this chapter, numerical determinations of these two parameters depend on groundwater abstraction from a well and the simultaneous records keeping of time and drawdown during an aquifer (pumping) test.



**Fig. 4.1** Transmissivity coefficient definitions



**Fig. 4.2** Saturation and unsaturation zones

### 4.3 Confined and Unconfined Aquifer Tests

An aquifer test is the record of drawdown with time in a main or observation or possibly in a group of adjacent wells. For the step drawdown test, the drawdown is generated in the main well by a sequence of constant discharges each with a specified time duration. Such tests are field experiments, and they are in absolute necessity to measure and evaluate the aquifer performance. The measurements do not yield the aquifer properties directly, but they are analyzed through different methodologies depending on the aquifer well types and subsequently the necessary interpretations come into view.

Rational and logical interpretations should precede the analyses. In the identification of aquifer parameters, most often the field data are plotted on a semi- or double-logarithmic paper, and theoretical type curves are matched mechanically for conventional calculations. Such an approach does not provide only bias or even irrelevant aquifer parameters, but hinders the expert view development

if the deviations from smooth type curves are not assessed carefully prior to mechanical theoretical curve matching. It is, therefore, necessary to consider the aquifer test data scatter primarily on ordinary, semi-logarithmic, or double-logarithmic papers through the filters of rational thinking and logic, which are referred to collectively as the “aquifer test philosophy.”

For philosophical evaluation of aquifer tests, it is necessary to have not only numerical values, but logical verbal statements concerning the basic features of the aquifer composition from general and structural geological points of view. One should keep in mind always that the mathematical formulations and models are based on a set of simplifying assumptions, which should be questioned prior to any aquifer test evaluation. If the assumptions are not judged as to their relationship with the actual conditions, then one cannot think philosophically. Although assumptions are necessary for rational approaches, they are restrictions to philosophical thinking, which considers the heterogeneity, anisotropy, and radial unsymmetrical flow. All scientific models have assumptions about the system based on certain

physical and geometrical features. It must not be forgotten that all theoretical type curves are analytical solutions, which simplify the system for solutions by a set of basic assumptions. Hence, the analytical solutions are approximations to the real situations and they provide only the general trend without local or instantaneous changes.

On the other hand, it is almost a fixed rule that in porous and many fracture media, the late time-drawdown data appears as a straight-line on a semi-logarithmic paper, and therefore, Jacob straight-line method is applied without any further consideration (Sen 2015). Such an operation implies philosophically that all straight-line appearances of late time data match the late time portion of the type curves such as the Theis (1935) type curve. This type curve has one trace only, and consequently, it should have corresponding single straight-line on a semi-logarithmic paper (Cooper and Jacob 1946). The question is then how can one use the Jacob approach for any straight-line occurrence on the semi-logarithmic paper?

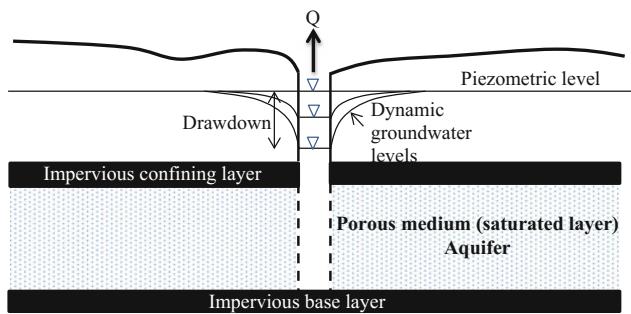
In order to gain experience for the philosophical interpretation of aquifer tests, three different paper types should be used by knowing the purpose of each one. The scatter of points on an ordinary paper helps to deduce basic rational and logical features even by non-experts. At least it provides the general trend, which can be depicted by any regression methodology, but it is not possible to deduce aquifer hydraulic parameters from such a trend, which gives only the shape of relationship quantitatively between time and drawdown or distance and drawdown. It is possible to see from an ordinary paper plot that the drawdown increases by time, and therefore, for large times it is expected to be approximately horizontal for confined and unconfined aquifers but horizontal for leaky aquifers. To magnify especially the late portion, the logarithm of the time variable is taken, and this leads to the semi-logarithmic paper usage for the evaluation of late time aquifer test data behaviors. Another advantage of using a semi-logarithmic paper is that most often late time data come along a straight-line. Logically, the more the slope of this line the smaller is the hydraulic conductivity (transmissivity) expectation. Hence, one can conclude verbally that the slope is inversely related to the transmissivity, which is defined as the hydraulic conductivity-saturation thickness product. Information that can be obtained from a straight-line is its intercept with one of the two axes and the slope.

It is the duty of aquifer test data analyst to correlate all pertinent information and provide a realistic picture of the groundwater system around the pumping well. It is safe to say that aquifer test data must be analyzed only after careful consideration of the geological composition around the well. The type curves or their mathematical expressions are merely means for accurate assessment of the aquifer parameters.

Prior to type curve matching or straight-line (Jacob) method application to available aquifer data, the question is “Is it not possible to deduce from the scatter of time-drawdown or distance-drawdown data some clues as for the type of aquifer medium whether it has fractured or karstic or porous (fine or coarse) material? If fractured has it double porosity or single fracture? Are fractures horizontal or vertical? Are they extensive or limited? With big or small aperture? Smooth or rough fracture surface?”

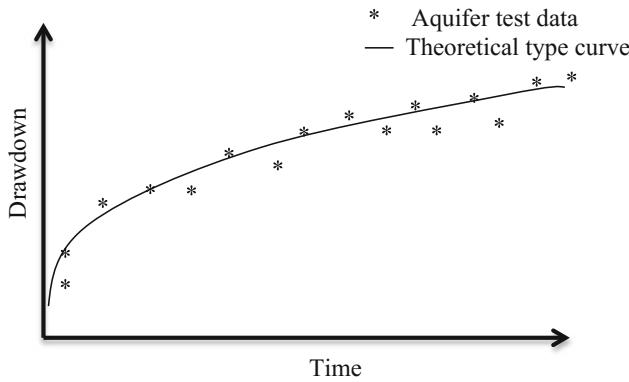
#### 4.3.1 Single Porous Media

The groundwater flow through sandy, gravelly, and silty, i.e., granular consolidated (sand stone) or unconsolidated (alluvium) medium has attracted many researchers since 1880s (Darcy 1856). Groundwater demand in arid zones and more importantly by the developments in petroleum engineering domain brought into the view well hydraulics research to identify water- or petroleum-bearing geological formations’ storage and transmissivity capacities. For this purpose, field pumping test is performed to measure drawdown variation records by time that are essential data for the water-bearing geological layer hydrogeological parameter identification (Fig. 4.3).



**Fig. 4.3** Confined aquifer

In practice, the water-saturated geological layer is referred to as the aquifer and the time-drawdown records are aquifer (pumping) test data. Logically, one deduces that as



**Fig. 4.4** Field data and theoretical curve

the pumping time increases so does the drawdown in the well and in the surrounding aquifer, the reflection of which seems in the form of scatter points around the most suitable theoretical curve as in Fig. 4.4.

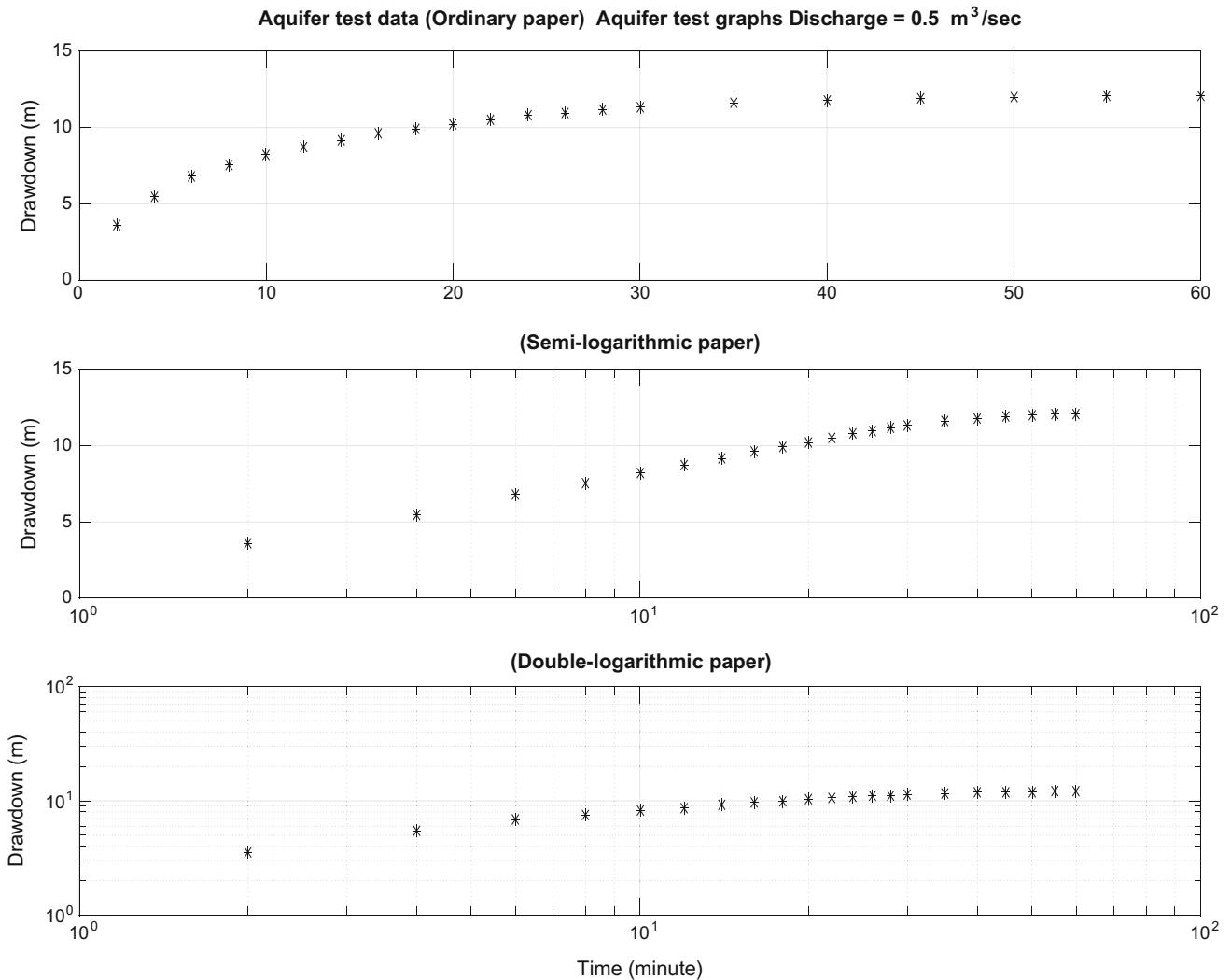
The theoretical curve has many different shapes depending on the type of aquifer material (porous, fractured, or karstic), well diameter (small or large) and penetration (fully or partially), groundwater flow law (linear or nonlinear, aquifer type (confined, unconfined, leaky). These curves are referred to as the “type curves,” and there are several textbooks in which mathematical derivation of type curves are given (Bear 1979; Freeze and Cherry 1984; Şen 1995, 2015).

The following MATLAB program plots aquifer test data on three different scales (ordinary, semi-logarithmic, and double-logarithmic).

```
function AquiferTestGraphs(T,D,Q,Title)
% This program is written by Zekai Şen on 4 April 2015
% z = Aquifer test time record
% d = Aquifer test drawdown record
% Q = Discharge
subplot(3,1,1)
plot(T,D,'k*')
ylabel('Drawdown (m)')
grid on; box on
title(['Aquifer test data (Ordinary paper) ' Title ' Discharge = ', num2str(Q) ' m^3/sec'])
subplot(3,1,2)
semilogx(T,D,'k*')
title('(Semi-logarithmic paper)')
ylabel('Drawdown (m)')
grid on; box on
subplot(3,1,3)
loglog(T,D,'k*')
xlabel('Time (minute)')
ylabel('Drawdown (m)')
grid on; box on
title('(Double-logarithmic paper)')
end
```

The following is a sample that comes out of this MATLAB program for aquifer test data plots on three papers (Fig. 4.5).

The following MATLAB program is similar to the previous one, but includes the aquifer parameter (transmissivity and storage coefficients' calculations).



**Fig. 4.5** Aquifer test data plots on three types of papers

```

function [Tmean Smean]=AquiferTestGraphsAndCalculations(T,D,Q,Title)
% This program is written on 4 April 2015 by Zekai Sen
% T = Aquifer test time sequence
% D = Drawdown(m)
% Q = Discharge m3/s
N=length(T); % Number of data
T=T*60; % Minute conversion to second
subplot(3,1,1)
plot(T,D,'k*')
ylabel('Drawdown (m)')
grid on; box on
title(['Aquifer test data (Ordinary paper) ' Baslik ' Discharge = ', num2str(Q)
' m^3/s'])
%title('Ordinary paper')
%legend('Aquifer test data')
subplot(3,1,2)
semilogx(T,D,'k*')
title('(Semi-logarithmic paper)')
ylabel('Drawdown (m)')
title>Title)
grid on; box on
% Aquifer parameter calculations
for i=2:N
    e(i-1)=(D(i,1)-D(i-1,1))/log(T(i,1)/T(i-1,1)); % Slope calculation
    T0(i-1)=T(i,1)*exp(-D(i,1)*log(T(i,1)/T(i-1,1))); % Horizontal axis cross point
    Tr(i-1)=4*Q/(4*pi*e(i-1)); % Transmissivity coefficient calculation
% NOTICE: Well radius value is necessary/If not available take as r=0.3 metre
% as assumption
r=0.3;
    St(i-1)=2.25*T0(i-1)*Tr(i-1)/(r*r); % Storage coefficient calculation
end
% The limits for first reliable calculations (more than 5% will be ignored)
k=0; %5 hata ile birbirine yakın olan eğimlerin sayacı
N2=N-2;
for i=1:N2
    re=100*abs((e(i+1)-e(i))/max(e(i+1),e(i))); % Relative error value
    if re <= 5 % 5% error limit is accepted
        k=k+1;
    else
        end
end
Tmean=60*60*24*sum(Tr(1:k))/k; % m2/gün'e çevrilmiş ortalama iletkenlik katsayısı
Smean=sum(St(1:k))/k; % Ortalama depolama katsayısı
%legend('Aquifer test data')
text(11,max(d)-5,['Transmissivity = ' num2str(Tmean) ' m^2/s'])
text(11,max(d)-15,['Storage = ' num2str(Smean)])
subplot(3,1,3)
loglog(T,D,'k*')
xlabel('Time (seconds)')
ylabel('Drawdown (m)')
title>Title)
grid on; box on
%legend('Aquifer test data')
title('(Double logarithmic paper)')
end

```

#### 4.3.1.1 Theis Method

The first type curve has been derived by Theis (1935) with the consideration from heat transfer approach as explained

by Carslaw and Jager (1947). This type curve is valid under a set of restrictive assumptions, which should be cared for prior to application to actual field data.

- (1) The aquifer is confined as shown in Fig. 4.3,
- (2) The groundwater flow is linear and abides with the Darcy law, which states that the groundwater velocity is directly and linearly proportional with the hydraulic gradient through a constant hydraulic conductivity (permeability) parameter,
- (3) The well penetrates the saturation thickness fully and its diameter is infinitesimally small,
- (4) The aquifer is extensively large, and there is no recharge from any source either externally or internally,
- (5) The pump discharge does not change throughout the pump test, but remains constant,
- (6) The aquifer material is homogeneous and isotropic.

Under these assumptions the groundwater flow abides with the mathematical expression toward a fully penetration well that can be found in aforementioned textbooks as,

$$W(u) = \int_{-\infty}^u \frac{e^{-x}}{x} dx \quad (4.1)$$

where  $W(u)$  is the well function and  $u$  is the dimensionless time factor,  $u$ , defined as,

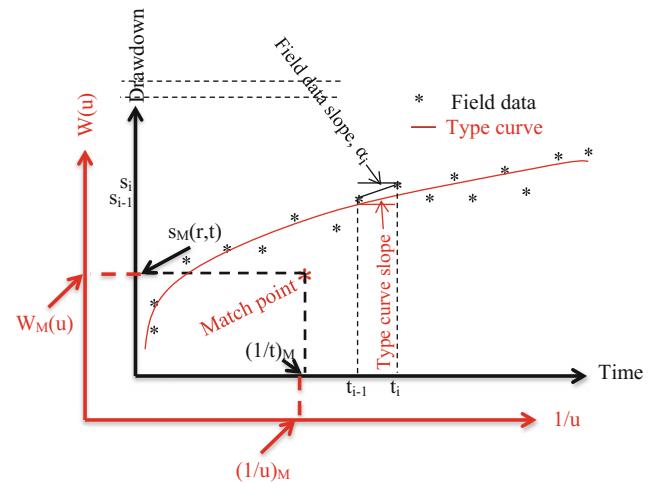
$$u = \frac{r^2 S}{4tT} \quad (4.2)$$

In this expression,  $r$  is the distance between the main and observation wells,  $S$  is the storage coefficient, and  $T$  is the transmissivity coefficient of the aquifer, and finally,  $t$  is the time duration for drawdown measurement,  $s$ . On the other hand, the explicit form of the well function is given as,

$$W(u) = \frac{4\pi T}{Q} s(r, t) \quad (4.3)$$

Herein,  $Q$  is the constant pump discharge and  $s(r, t)$  is the drawdown amount at radial distance,  $r$ , and at time,  $t$ . In Fig. 4.6, a representative field data scatter is shown with a theoretical type curve both on a double-logarithmic axes scale.

For the application of the procedure presented in this figure, the execution of the following steps is necessary.



**Fig. 4.6** Type curve and field data matching

- (1) Plot the scatter of field data (time-drawdown) on a double-logarithmic paper,
- (2) Plot the theoretical type curve on a separate double-logarithmic paper  $[1/u - W(u)]$  with the same scale,
- (3) Overlap these two double-logarithmic papers and slide on each other until the best visual match is obtained between the type curve and the scatter points. During the slide, the axes must always remain parallel to each other,
- (4) Take arbitrarily any point that is within the common area of the two papers, which is called as the “match point,”
- (5) Read two values from the horizontal and vertical axes, and hence, there are four numerical values at hand, which are  $(1/t)_M$ ,  $(1/u)_M$ ,  $W_M(u)$ , and  $s_M(r, t)$  according to the match point,
- (6) Substitution of these values into convenient equations [Eqs. (4.2) and (4.3)] yields the aquifer parameter numerical values for storage and transmissivity coefficients.

The above-explained procedure remains applicable for any kind of type curve that are available in the hydrogeology or petroleum geology works. The MATLAB program for the Theis type curve is given below.

```

function [UF, WF] = Theis(N)
T1=0.00000179;
S1=T1;
V1=5.0;
U1=100.0;
for I=1:8
    V1=V1/10.0;
    U1=U1/10.0;
    XU=U1;
    for J=1:18
        U=U1-J*V1;
        XL=U;
        H=(XU-XL)/N;
        FV1=T1;
        for K=1:N
            X=XL+K*H;
            FV2=exp(-X)/X;
            S1=S1+0.5*H*(FV1+FV2);
            FV1=FV2;
        end
        UT(I,J)=U;
        WT(I,J)=S1;
        T1=S1;
        XU=XL;
    % WT(I,J) = -0.577216-log10(U)+U-(U*U)./(2*factorial(2))+ (U*U*U)./(3*factorial(3));
    end
end
k=0;
for i=1:8
    for j=1:18
        k=k+1;
        UF(k)=UT(i,j);
        WF(k)=WT(i,j);
    end
end
figure
loglog(1./UF,WF,'k','LineWidth',2)
hold on
title('Confined aquifer well functions - Theis')
xlabel('Inverse dimensionless time factor (1/u)')
ylabel('Well function W(u)')
grid on
box on
end

```

Theis type curve is given in Fig. 4.7 as dimensionless time factor versus the well function on the vertical axis that comes out of the MATLAB program.

#### 4.3.1.2 Large Diameter Well Type Curves

Similar to infinitesimally small diameter type curves by Theis (1935), Papadopoulos and Cooper (1967) provided

large diameter well type curves under the same set of assumptions in addition to the well diameter not infinitesimally small. The first time in this book, the following MATLAB program is given for large diameter main and small diameter observation wells' type curves in the well vicinity.

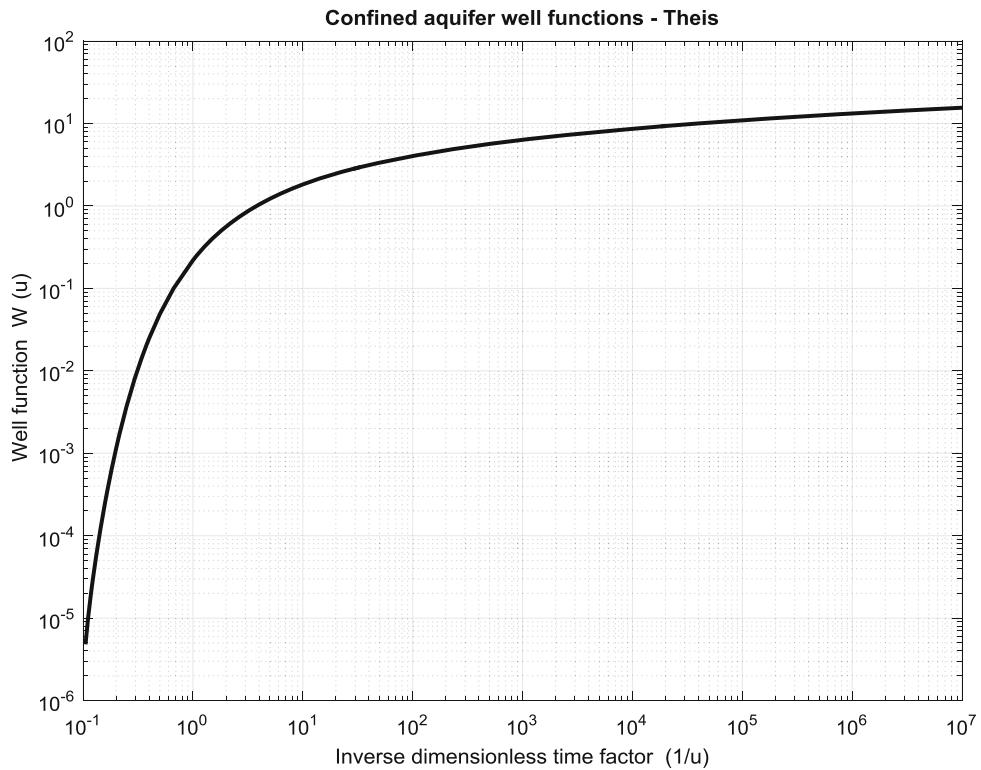
```

function [UF,WF] = LargeDiameterWellMainAndObservationTypeCurves(N,r)
% This program calculates numerically type curves in large diameter wells for r = 1
% or in observation wells around large diameter wells for r < 1.
%
% REFERENCES
% Papadopoulos, I.S., and Cooper, H.H., (1967). Drawdown in a well of large
% diameter. Water Resources Research, Vol. 3, No.1: 241-244.
% Sen, Z., (2015). Practical and Applied Hydrogeology. Elsevier,
% Amsterdam, 406 pages
%
S=[0.1 0.01 0.001 0.0001 0.00001];
r2=r*r;
for L=1:5
    W1=0.00000179;
    V1=5;
    U1=100.0;
    for I=1:8
        V1=V1/10.0;
        U1=U1/10.0;
        XU=U1;
        for J=1:18
            U=U1-J*V1;
            XL=U;
            H=(XU-XL)/N;
            for K=1:N
                X=XL+K*H;
                F1=exp(-X)/X;
                F2=1+(1/S(L))*r2*X*exp(-r2*X);
                W1=W1+F1*H/F2;
            end
            UT(I,J)=U;
            WT(I,J)=W1;
            XU=XL;
        end
    end
    k=0;
    for i=1:8
        for j=1:18
            k=k+1;
            UF(L,k)=UT(i,j);
            WF(L,k)=WT(i,j);
        end
    end
end
figure
for i=1:5
    loglog(1./UF(i,19:126),WF(i,19:126),'LineWidth',2)
    hold on
    title('Confined aquifer large diameter well type curves')
    xlabel('Inverse dimensionless time factor (1/U_L)')
    ylabel('Well function W(U_L,S)')
    grid on
    box on
    legend('Storage Coeff. = 0.1','Storage Coeff. = 0.01','Storage Coeff. =

```

In Fig. 4.8a, the main well type curves are given for a set of storage coefficient values in addition to observation wells at various distances from the main well in Fig. 4.7b-d. The parameter  $r$  is defined as the ratio of the main well diameter to observation well distance from the main well, and hence, as the distance increases from the main well the  $r$  value becomes smaller. Furthermore, as the observation

well distance increases the type curves approach to each other and at large distances the large diameter well effect decreases on the observation well type curves. For this reason, at large distances the type curve (see Fig. 4.8d) overlaps with the Theis type curve in Fig. 4.7. The reader can compare these two figures and appreciate the result.

**Fig. 4.7** Theis type curve

#### 4.3.1.3 Şen Slope Method

This method does not need a type curve matching procedure, but the field data slope match to type curve slope is sufficient to calculate aquifer parameters (Şen 1986). The aquifer parameter estimations can be achieved after the second time-drawdown measurement, and then the same slope calculation procedure continues with the coming of successive measurements. As can be seen from Fig. 4.6, matching a type curve is equivalent to match the slopes of the field and type curve. It is obvious that the field data slopes are almost random to a certain extent, and therefore, not single storativity and transmissivity coefficient values are obtained; instead, sequences of these parameters are calculable for each time interval. If the pump test data number is  $n$ , then there will be  $n - 1$  parameter values. In this manner, the aquifer heterogeneity is reflected in the parameter values. The aquifer parameters are dependent on heterogeneous geological composition of the aquifer. In the conventional type curve matching procedures, the aquifer is assumed to represent an equivalent homogeneous and isotropic geological medium, and consequently, the parameters are expected to be temporally and spatially invariable constants. The theoretical analytic slope,  $\alpha$ , expression for the Theis curve on a double-logarithmic paper can be calculated as (Şen 1986),

$$\alpha = -\frac{e^{-u}}{W(u)} \quad (4.4)$$

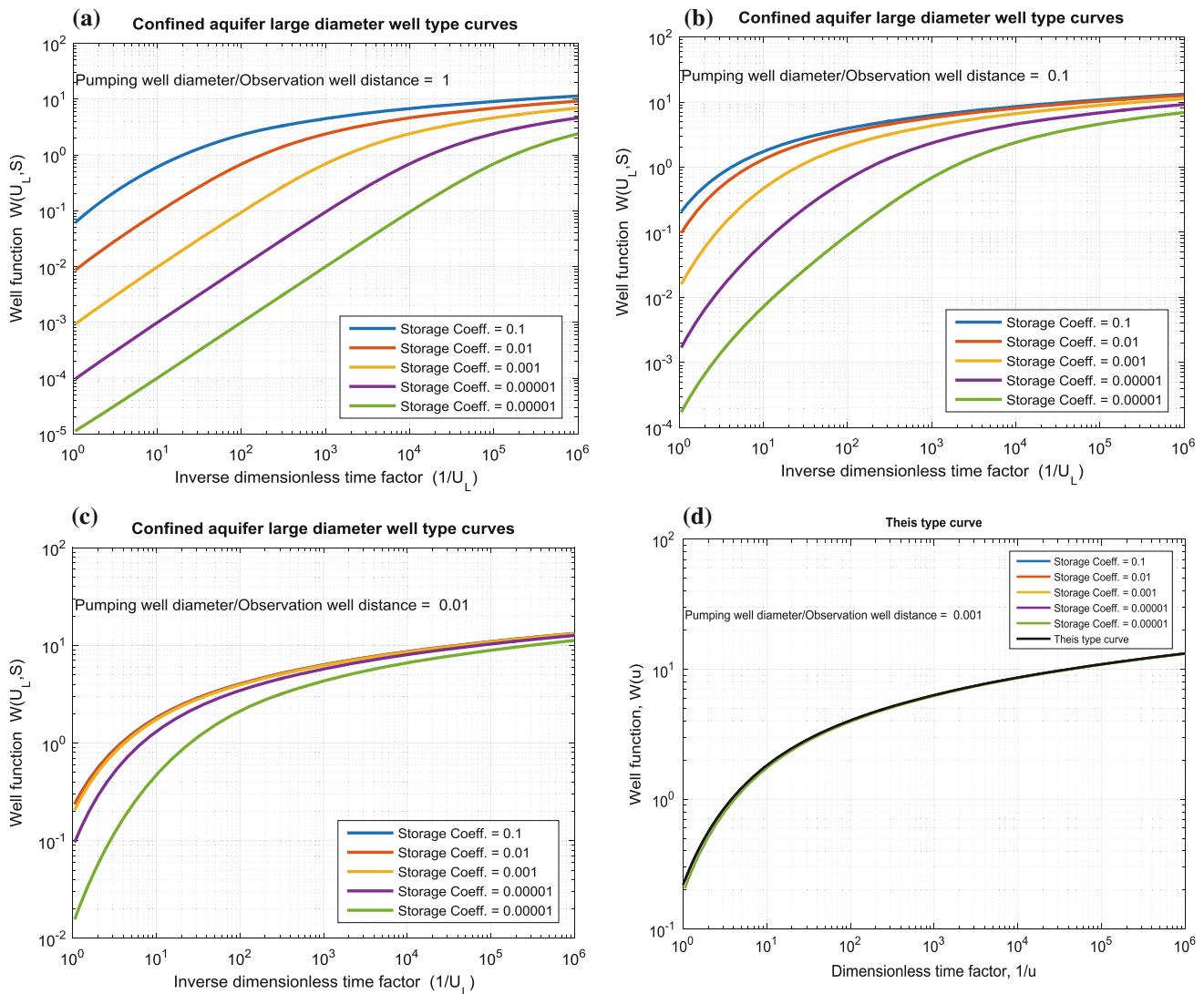
This expression helps to convert the Theis type curve table into slope values as presented in Table 4.1.

By considering this table, the processing of the aquifer test data can be achieved through the following steps without type curve matching.

- After the second time-drawdown measurement, calculate the slope between the two successive points in the double-logarithmic scale as,  $\alpha_i = \ln(s_i/s_{i-1}) / \ln(t_{i-1}/t_i)$  (see Fig. 4.6) where  $i = 2, 3, \dots, n$  and  $n$  is the number of drawdown records,
- Find the  $u$  value corresponding to this slope from Table 4.1 (if needed interpolate),
- Knowing the slope and the  $u$  value calculate the well function value from Eq. (4.4) as,

$$W_i(u) = \frac{e^{-u_i}}{\alpha_i} \quad (4.5)$$

- Calculate local  $T$  and  $S$  values from Eqs. (4.2) and (4.3), respectively,



**Fig. 4.8** Large diameter well vicinity observation well type curve, **a**  $r = 1.0$ , **b**  $r = 0.1$ , **c**  $r = 0.01$ , **d**  $r = 0.001$

- (e) Repeat the previous steps with the next time-drawdown record. Finally, sequences of estimation are obtained for each aquifer parameter.

The slope method matches every successive pair of drawdown measurement slope, i.e., drawdown per time, with

a part of the type curve slope that is given in Fig. 4.9, which is the reflection of slopes in Table 4.1. The basic software in MATLAB is given in the following box. The software is similar to the Theis type curve software except for the slope calculation.

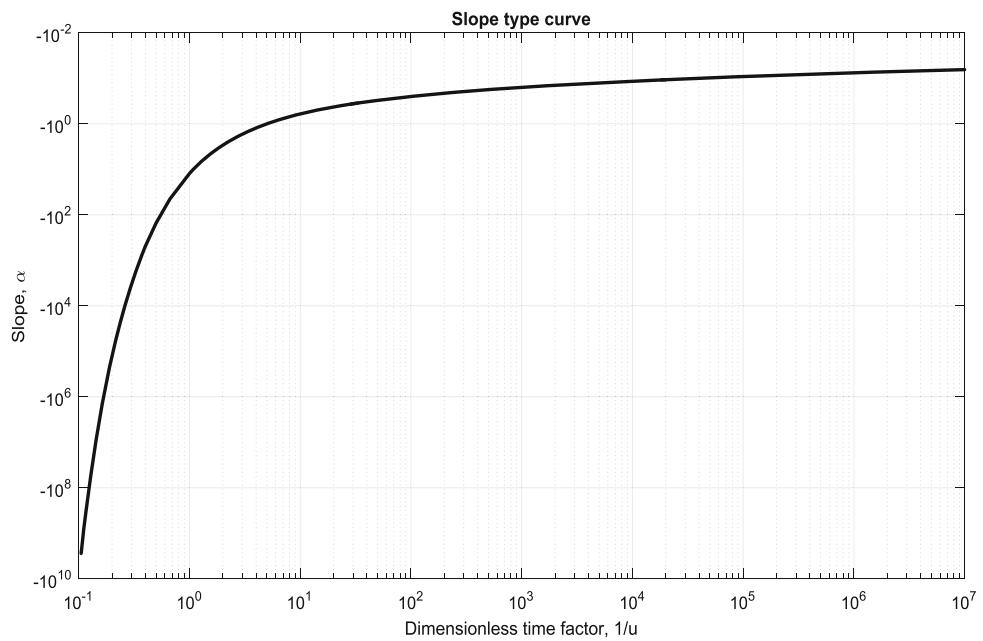
```

function [UF, WF, Slope] = SlopeTypeCurve(N)
T1=0.00000179;
S1=T1;
V1=5.0;
U1=100.0;
for I=1:8
    V1=V1/10.0;
    U1=U1/10.0;
    XU=U1;
    for J=1:18
        U=U1-J*V1;
        XL=U;
        H=(XU-XL)/N;
        FV1=T1;
        for K=1:N
            X=XL+K*H;
            FV2=exp(-X)/X;
            S1=S1+0.5*H*(FV1+FV2);
            FV1=FV2;
        end
        UT(I,J)=U;
        WT(I,J)=S1;
        T1=S1;
        XU=XL;
%           WT(I,J) = -0.577216-log10(U)+U-
%(U*U)./(2*factorial(2))+(U*U*U)./(3*factorial(3));
    end
end
k=0;
for i=1:8
    for j=1:18
        k=k+1;
        UF(k)=UT(i,j);
        WF(k)=WT(i,j);
    end
end
Slope=-exp(UF)./WF;
loglog(1./UF,Slope,'k','LineWidth',2)
title('Slope type curve')
xlabel('Dimensionless time factor, 1/u')
ylabel('Slope, \alpha')
grid on
end

```

**Table 4.1** Type curve slopes

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
$\times 10^0$	-1.6798	-2.7619	-3.8298	-4.8199	-6.1254	-6.8854	-7.5990	-8.8279	-10.2841
$\times 10^{-1}$	-0.4971	-0.6711	-0.8141	-0.9576	-1.0831	-1.2197	-1.3421	-1.4494	-1.5637
$\times 10^{-2}$	-0.2451	-0.2926	-0.3278	-0.3585	-0.3851	-0.4095	-0.4337	-0.4547	-0.4760
$\times 10^{-3}$	-0.1578	-0.2769	-0.1906	-0.2014	-0.2104	-0.2189	-0.2262	-0.2329	-0.2394
$\times 10^{-4}$	-0.1159	-0.1259	-0.1322	-0.1379	-0.1424	-0.1461	-0.1494	-0.1525	-0.1551
$\times 10^{-5}$	-0.0914	-0.0976	-0.1016	-0.1047	-0.1072	-0.2094	-0.1112	-0.1128	-0.1144
$\times 10^{-6}$	-0.0755	-0.0797	-0.824	-0.0844	-0.0859	-0.0873	-0.0886	-0.0896	-0.0906
$\times 10^{-7}$	-0.0653	-0.0673	-0.0692	-0.0707	-0.0728	-0.0729	-0.0735	-0.0743	-0.0745
$\times 10^{-8}$	-0.0560	-0.0583	-0.0597	-0.0607	-0.0616	-0.0623	-0.0629	-0.0634	-0.0640
$\times 10^{-9}$	-0.0496	-0.0514	-0.0524	-0.0533	-0.0539	-0.0545	-0.0549	-0.0553	-0.0557
$\times 10^{-10}$	-0.0445	-0.0459	-0.0468	-0.0475	-0.0480	-0.0484	-0.0488	-0.0491	-0.0494

**Fig. 4.9** Slope type curve

As the output of this MATLAB program comes the slope type curve conversion from the Theis confined aquifer type curve as in Fig. 4.9. It is possible to obtain the theoretical slope value corresponding to any dimensionless time factor,  $u$ , value.

#### 4.3.1.4 Leaky Aquifer Test

A leaky aquifer has three layers from top to bottom, unconfined saturation layer, semi-confined leaky layer, and confined saturation layer. There is water exchange between the unconfined and confined aquifers through the leaky layer (Fig. 4.10).

The leaky aquifer has a different type curve in which apart from the storage and transmissivity coefficients additionally there is leakage factor. The leaky aquifer type curve expressions are suggested by Hantush (1956) for infinitesimally small diameter (theoretically zero well radius) wells. The same set of assumptions is valid as for the Theis type

curve (Sect. 4.3.1.1). The following mathematical expressions are enough for the determination of the three leaky aquifer layer parameters. The general form of the leaky aquifer well function expression is given as,

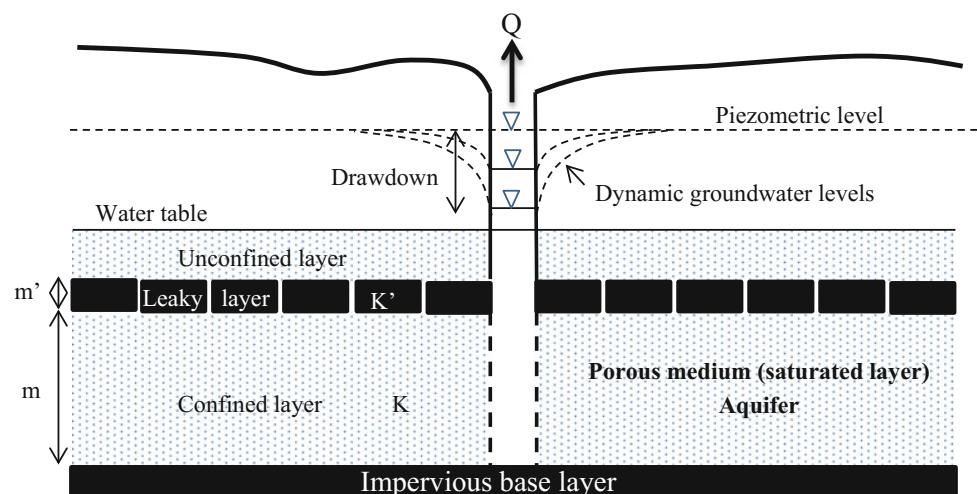
$$W(u, \frac{r}{L}) = \int_{-\infty}^u \frac{1}{x} \exp\left(x - \frac{r^2}{4L^2x}\right) dx \quad (4.6)$$

Similar expressions are valid for storage and transmissivity coefficients calculations according to the following two equations.

$$W(u, \frac{r}{L}) = \frac{4\pi T}{Q} s(r, t) \quad (4.7)$$

where dimensionless time function is,

$$u = \frac{r^2 S}{4tT} \quad (4.8)$$

**Fig. 4.10** Leaky layer

Additionally, the leakage factor is also given as,

$$L = \sqrt{\frac{m'mK}{K'}} \quad (4.9)$$

In this expression,  $m$  and  $m'$  are the confined and leaky layer thicknesses;  $K$  and  $K'$  are the hydraulic conductivities of the confined and leaky layers, respectively. The leaky aquifer type curves can be obtained from the MATLAB program given below for infinitesimally small well radius cases.

```

function [UF, WF] = HantushTypeCurves(N)
% This program is for leaky aquifer type curve calculation with the assumption that the well
% diameter is equal to zero.
% The resultant type curves are valid for observation wells only.
rb=[0.01 0.05 0.15 0.30 0.50 0.70 1.00 2.00 2.50];
for L=1:9
    T1=0.00000179;
    S1=T1;
    V1=5.0;
    U1=100.0;
    for I=1:8
        V1=V1/10.0;
        U1=U1/10.0;
        XU=U1;
        for J=1:18
            U=U1-J*V1;
            XL=U;
            H=(XU-XL)/N;
            FV1=T1;
            for K=1:N
                X=XL+K*H;
                FV2=exp(-X-rb(L)*rb(L)/X);
                S1=S1+0.5*H*(FV1+FV2);
                FV1=FV2;
            end
            UT(I,J)=U;
            WT(I,J)=S1;
            T1=S1;
            XU=XL;
        end
    end
    k=0;
    for i=1:8
        for j=1:18
            k=k+1;
            UF(L,k)=UT(i,j);
            WF(L,k)=WT(i,j);
        end
    end
end
for i=1:9
    loglog(1./UF(i,:),WF(i,:),'LineWidth',2)
    hold on
    title('Hantush leaky aquifer type curves')
    xlabel('Inverse dimensionless time factor (1/u)')
    ylabel('Well function W(u,\beta)')
    grid on
    box on
    legend('\beta = 0.01', '\beta = 0.05', '\beta = 0.15', '\beta = 0.30', ...
        '\beta = 0.50', '\beta = 0.70', '\beta = 1.00', '\beta = 1.50', '\beta = 2.00', 'Location', 'SouthEast')
end

```

This software yields the set of leaky aquifer type curves for each leakage factor as in Fig. 4.11.

In this book, first time large diameter well leaky aquifer type curve software is presented in the following box. Herein, like Sect. 4.3.1.2,  $r$  indicates the ratio of the large

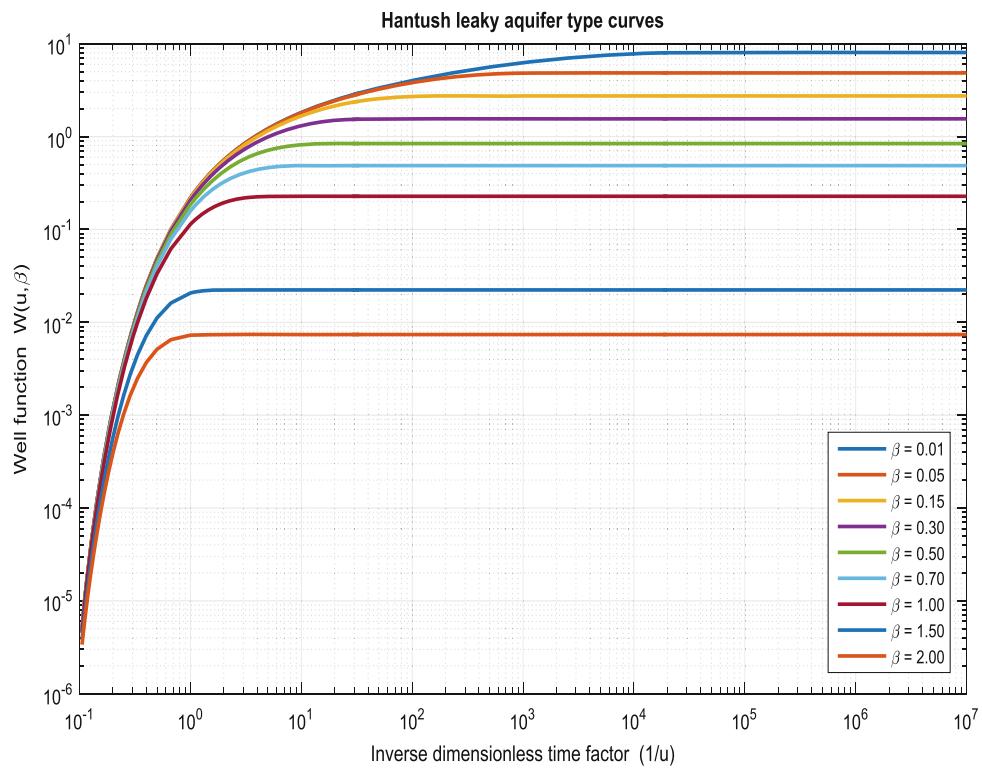
diameter well to the observation well distance. In the large diameter well case  $r = r_w = 1$ , where  $r_w$  is the large diameter well radius. For any observation well distance,  $d$ , more than  $r_w$ , the distance ratio is defined as  $r = r_w/d$  as a number less than 1.

```

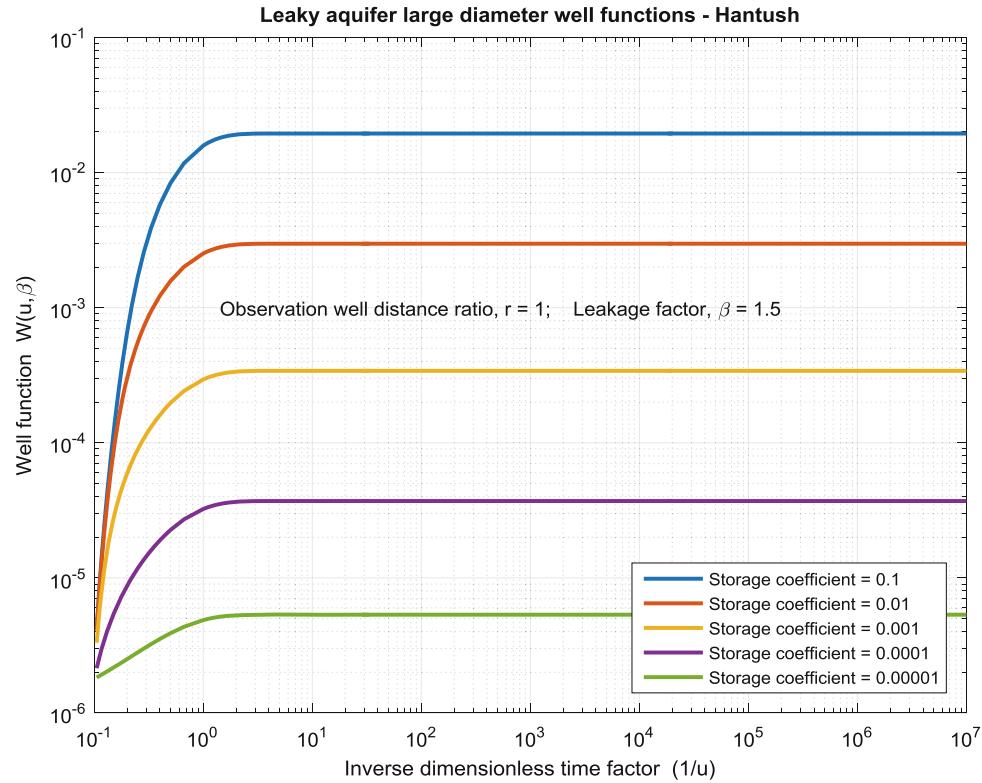
function [UF,WF] = LargeDiameterWellHantushTypeCurves(N,r,rb)
% This program calculates numerically type curves in large diameter wells in leaky aquifers %
% for r = 1 for the main well or in observation wells around large diameter wells for r < 1.
% rb = leakage factor
%
% REFERENCE: Sen, Z., (1996). Volumetric leaky aquifer theory and type
% straight lines. ASCE, Journal of Hydrologic Engineering, Vol. 22, No.5: 272-280.
%
S=[0.1 0.01 0.001 0.0001 0.00001];
r2=r*r;
rb2=rb*rb;
for L=1:5
    W1=0.00000179;
    V1=5.0;
    U1=100.0;
    for I=1:8
        V1=V1/10.0;
        U1=U1/10.0;
        XU=U1;
        for J=1:18
            U=U1-J*V1;
            XL=U;
            H=(XU-XL)/N;
            for K=1:N
                X=XL+K*H;
                F1=exp(-X-rb2/X);
                F2=1+(1/S(L))*r2*X*exp(-r2*X);
                W1=W1+F1*H/F2;
            end
            UT(I,J)=U;
            WT(I,J)=W1;
            XU=XL;
        end
    end
    k=0;
    for i=1:8
        for j=1:18
            k=k+1;
            UF(L,k)=UT(i,j);
            WF(L,k)=WT(i,j);
        end
    end
end
figure
for i=1:5
    loglog(1./UF(i,:),WF(i,:),'LineWidth',2)
    hold on
end
title('Leaky aquifer large diameter well functions - Hantush')
xlabel('Inverse dimensionless time factor (1/u)')
ylabel('Well function W(u,\beta)')
grid on
box on
legend('Storage coefficient = 0.1','Storage coefficient = 0.01',...
    'Storage coefficient = 0.001','Storage coefficient = 0.0001',...
    'Storage coefficient = 0.00001')
text(min(1./UF(1,:)),2*max(WF(1,:)),[' Observation well distance ratio, r = ...
    ,num2str(r),' Leakage factor, \beta = ', num2str(rb)])
end

```

**Fig. 4.11** Leaky aquifer type curves



**Fig. 4.12** Large diameter well type curves in a leaky aquifer



The execution of the software in the above box for the large diameter well itself ( $r = 1$ ) leads to the consequent type curves given in Fig. 4.12 for leakage factor 1.5.

One can obtain type curves for any observation well around the large diameter main well within a leaky aquifer again by the execution of the software in the above box, but

with distance ratio less than 1. For instance, in Fig. 4.13 three observation well type curves are given for this case for distance ratio as  $r = 0.1$ ,  $r = 0.01$ , and  $r = 0.0001$ .

A close inspection of these type curves indicates to the following interpretations for large diameter main well observation type curves in leaky aquifers.

- (1) At large times, all the type curves take a horizontal line position indicating the leaky characteristic of the aquifer (see Fig. 4.13a),
- (2) As the distance ratio increases, i.e., as the observation well distance from the main well increases, the type curves at different storage coefficients come closer to each other with increasing well function values (compare Fig. 4.13a, b),

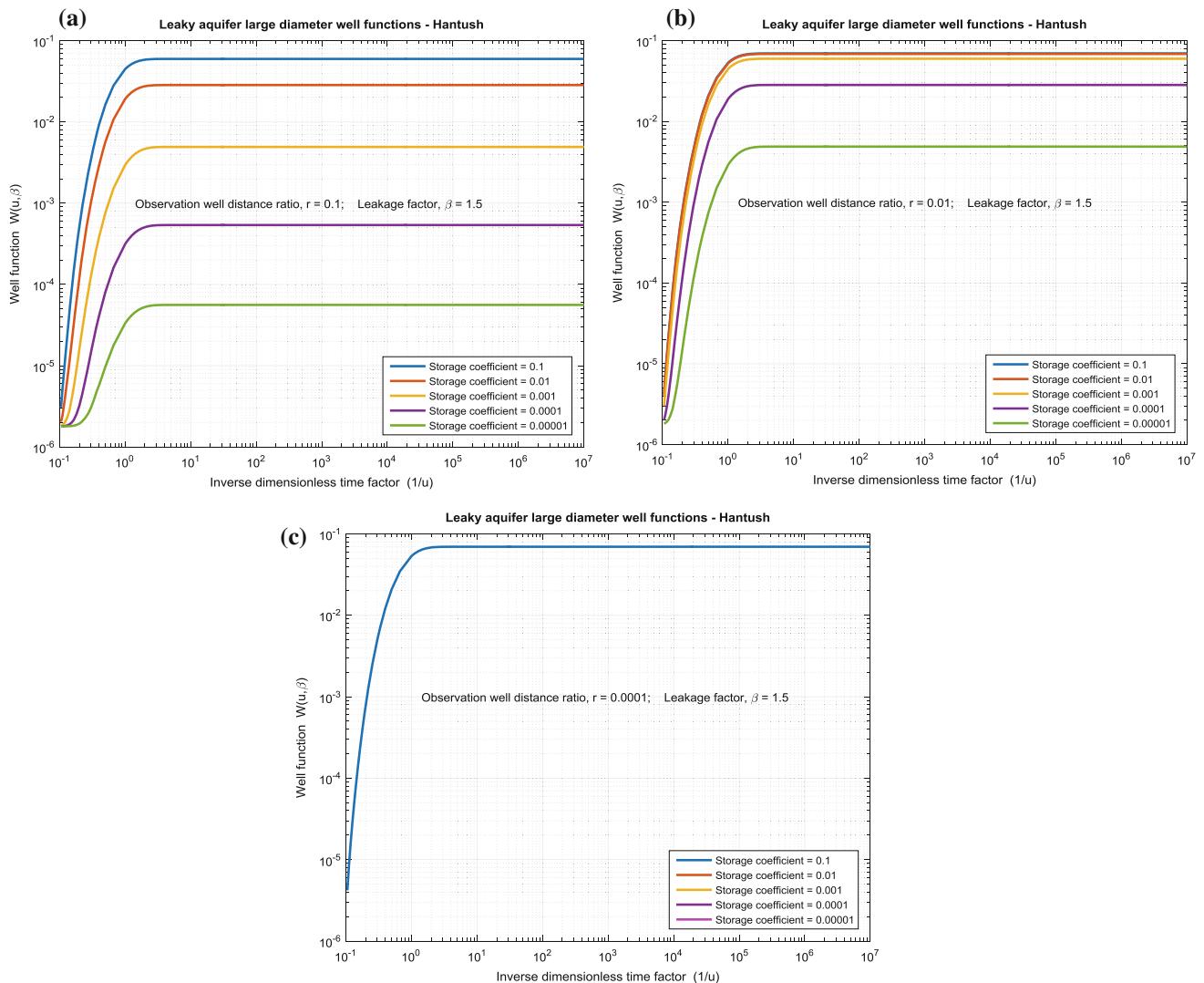
- (3) At very big distance ratios all the previous type curves collapse on to a single one as in Fig. 4.13c.

At late times, dimensionless time factor,  $u$ , goes to zero, and hence, type curves take the form of straight-lines on semi-logarithmic paper. The straight-line type expression is given by Sen (1985) as follows.

$$W(u, r/L, \eta) = \left(1 - \frac{1}{\eta}\right) K_0 \left(\frac{r}{L} \sqrt{\frac{\eta}{1-\eta}}\right) - \frac{0.5772}{\eta} - \frac{2.3}{\eta} \log(u) \quad (4.10)$$

This shows a straight-line relationship between  $W(u, r/B, \eta)$  and  $u$  on a semi-logarithmic paper. The following MATLAB program solves this straight-line mathematical expression in the last equation numerically.

```
function [UF, WF] = LeakyAquiferTypeLine(eta)
% This software calculates type straight-lines in a leaky aquifer
% rL is the leakage factor
% eta is defined as (1+The ratio of the storage coefficient of unpumped overlying unconfined
% aquifer to the underlying leaky pumped aquifer), i.e. (eta=1+S'/S)
%
% REFERENCE: Sen, Z., (1996). Volumetric leaky aquifer theory and type
% straight lines. ASCE, Journal of Hydrologic Engineering, Vol. 22, No.5: 272-280.
%
rL=[0.001 0.01 0.1 1];
figure
for L=1:4
c=rL(L)*sqrt(eta/(eta-1));
V1=5.0;
U1=100.0;
for I=1:8
V1=V1/10.0;
U1=U1/10.0;
for J=1:18
U=U1-J*V1;
UT(I,J)=U;
WT(I,J)=(1-1/eta)*besselk(0,c)-0.5772/eta-(2.3/eta)*log10(U);
end
end
k=0;
for i=1:8
for j=1:18
k=k+1;
UF(k)=UT(i,j);
WF(k)=WT(i,j);
end
end
semilogx(1./UF,WF,'LineWidth',2)
hold on
end
title('Leaky aquifer type straight-lines')
xlabel('Inverse dimensionless time factor (1/u)')
ylabel('Well function W(u,r/L,\eta)')
grid on
box on
text(min(1./UF),max(WF),['\eta = ',num2str(eta),' c = ',num2str(c)])
legend('r/L=0.001','r/L=0.01','r/L=0.1','r/L=1.0','Location','SouthEast')
end
```



**Fig. 4.13** Leaky aquifer observation well type curves

The execution of this software with various leakage coefficients yields to type straight-lines for the leaky aquifers and a sample example is given in Fig. 4.14 for  $\eta = 10$ .

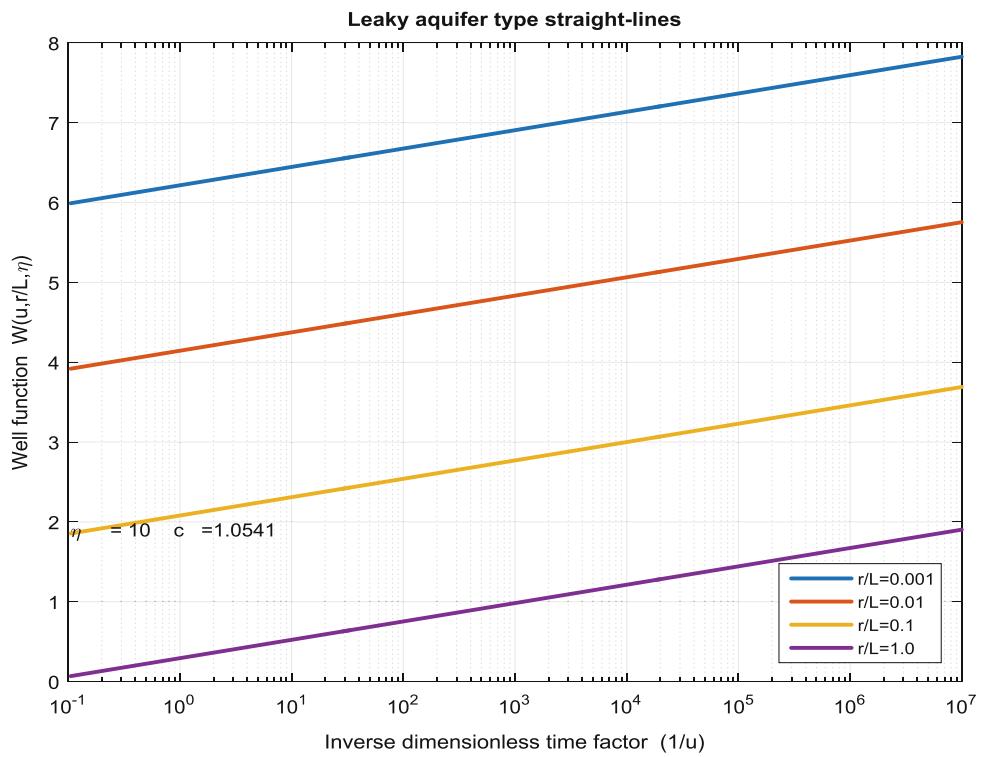
#### 4.4 Fractured Aquifer Test

Geometry of natural fractures is complex and difficult for exact solutions. Idealizations through a set of assumptions are necessary prior to any treatment of the flow in naturally fractured rock media. A fractured aquifer is regarded as a

result of a certain pattern of fractures or the configuration of a few finitely or infinitely extended fractures. So far in the literature three simplification procedures are presented (Şen 1986b).

- (a) Barenblatt et al. (1960) conceptualized the fractured reservoir as two coexisting and interacting media, namely primary porosity of blocks with low permeability with high storativity and a medium of secondary porosity of fractures with high permeability, but low storativity. Each medium acts as a classical porous

**Fig. 4.14** Leaky aquifer type straight-line



medium; however, their interaction is different than any homogeneous and isotropic porous medium concept due to the hydraulic interaction between them,

- (b) Regular Euclidean block geometry is assumed either in the forms of equal parallelepipeds surrounded by orthogonal fracture pattern or as alternate horizontal layer successions with significantly different hydraulic conductivities. Individual fractures are described by various parameters such as aperture size, roughness, orientation, dipping, frequency, and hydraulic radius. This is referred to “regular double porosity” model and used by Warren and Root (1963) and Kazemi (1969),
- (c) Fictitious continuum assumption is equivalent to actual fractured medium. It is called as “equivalent double

porosity” model (Sen 1995). Snow (1965) assumed that the fractures have infinite length with an equivalent porous medium permeability.

Even after these conceptual idealization and simplifications, the theoretical models are mathematically complex and lead to complicated type curve expressions with several parameters. However, if the type curves are available then the same matching procedure is applicable as explained in the previous sections.

The MATLAB software in the following box provides a single horizontal fracture type curve in a confined aquifer in case of an infinitesimally small diameter main well.

```

function [UF,WF] = horfra(N,beta)
% This program calculates the type curve of horizontal rough fracture
% beta = Fracture roughness coefficient
T1=exp(-10/beta)/108*beta;
S1=T1;
V1=5.0;
U1=100.0;
for I=1:8
    V1=V1/10.0;
    U1=U1/10.0;
    XU=U1;
    for J=1:18
        U=U1-J*V1;
        XL=U;
        H=(XU-XL)/N;
        FV1=T1;
        for K=1:N
            X=XL+K*H;
            FV2=exp(-X/beta)/(beta*X);
            S1=S1+0.5*H*(FV1+FV2);
            FV1=FV2;
        end
        UT(I,J)=U;
        WT(I,J)=S1;
        T1=S1;
        XU=XL;
    end
end
k=0;
for i=1:8
    for j=1:18
        k=k+1;
        UF(k)=UT(i,j);
        WF(k)=WT(i,j);
    end
end
figure
loglog(1./UF,WF,'k','LineWidth',2)
hold on
title('Confined aquifer horizontal fracture well functions - Zekai')
xlabel('Inverse dimensionless time factor (1/u)')
ylabel('Well function W(u)')
text(min(1./UF),2*max(WF),['\beta = ' num2str(beta)]);
grid on
box on
end

```

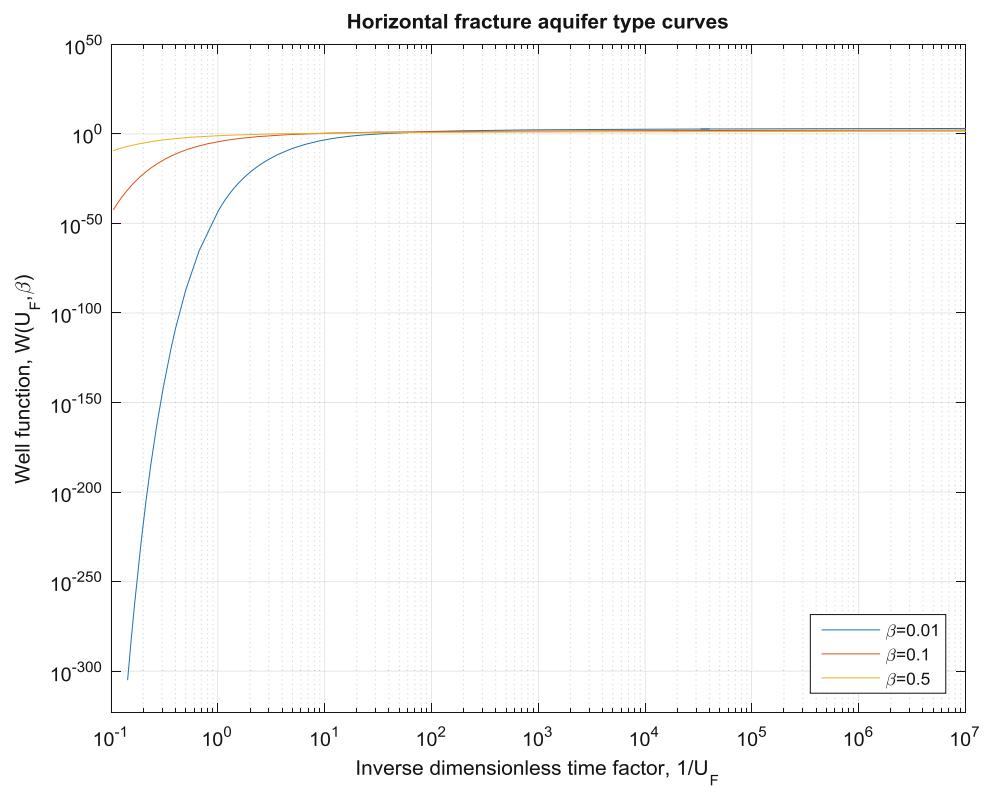
The horizontal fracture aquifer type curves are shown for dimensionless fracture aperture factors,  $\beta = 0.01; 0.1; 0.5$ , in Fig. 4.15.

On the other hand, by use of the same MATLAB program, it is possible to obtain type curves for

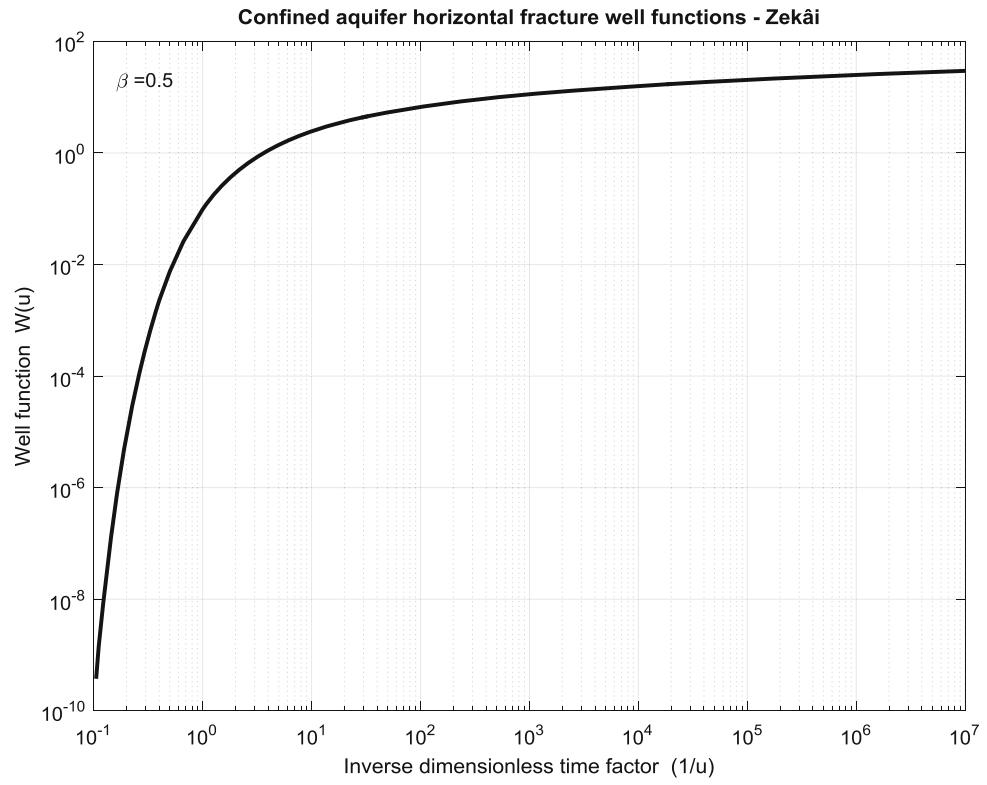
dimensionless fracture factors for which a sample is presented in Fig. 4.16.

It is also possible to consider a horizontal rough fracture aquifer with a finite aperture value. In this case, the type curves can be obtained by the following MATLAB software.

**Fig. 4.15** Various dimensionless horizontal fracture factor type curves



**Fig. 4.16** Horizontal fracture type curves for dimensionless horizontal fracture factor 0.5



```

function [UF,WF] = horfrafindif(N,r)
% This program calculates numerically type curves of flow through a rough
% and horizontal fracture (well storage is not considered)
% r is the dimensionless fracture aperture factor
% N is the number if finite difference intervals
for L=1:1
    W1=exp(-10/r)/10;
    V1=5.0;
    U1=100.0;
    for I=1:8
        V1=V1/10.0;
        U1=U1/10.0;
        XU=U1;
        for J=1:18
            U=U1-J*V1;
            XL=U;
            H=(XU-XL)/N;
            for K=1:N
                X=XL+K*H;
                F1=exp(-X/r)/X;
                F2=1;
                W1=W1+F1*H/F2;
            end
            UT(I,J)=U;
            WT(I,J)=W1/r;
            XU=XL;
        end
    end
    k=0;
    for i=1:8
        for j=1:18
            k=k+1;
            UF(L,k)=UT(i,j);
            WF(L,k)=WT(i,j);
        end
    end
end
figure
loglog(1./UF,WF,'k','LineWidth',2)
hold on
title('Horizontal fracture large diameter well functions- Zekai without well
storage')
xlabel('Inverse dimensionless time factor (1/u)')
ylabel('Well function W(u)')
text(min(1./UF),2*max(WF),['Dimensionless fracture aperture =' num2str(r)]);
grid on
box on
end

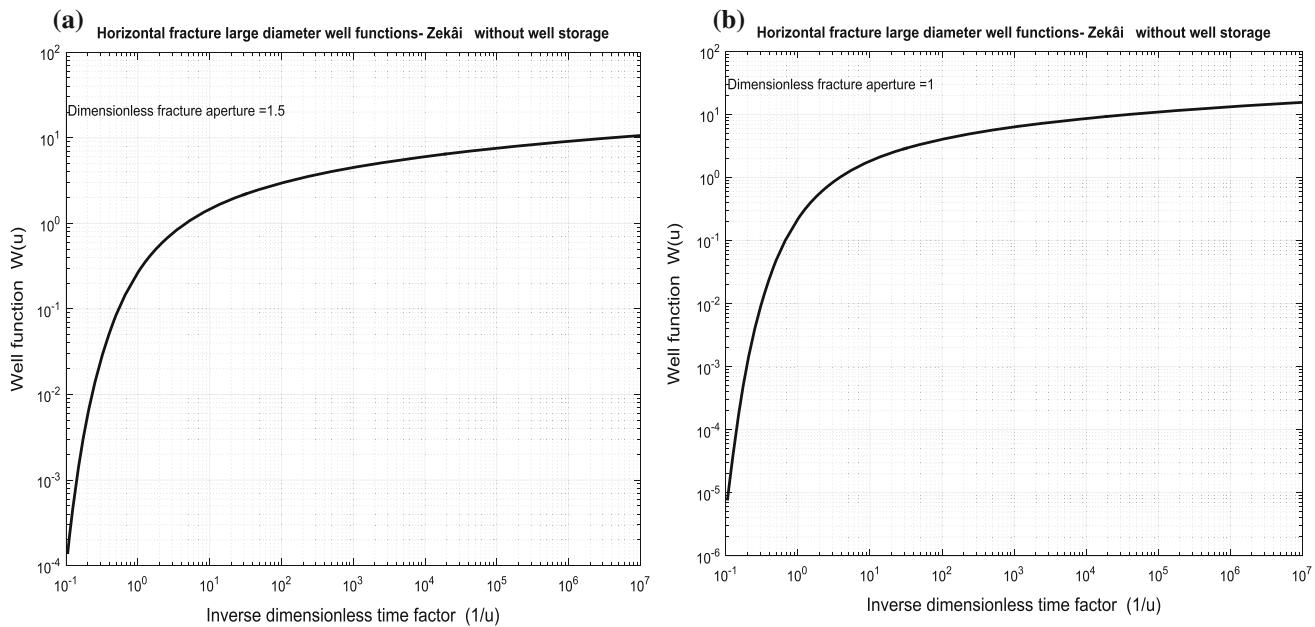
```

The output of this MATLAB program yields to the rough horizontal fracture finite dimensionless fracture aperture factor type curves as in Fig. 4.17 for  $r = 1.0$  and  $r = 1.5$ .

## 4.5 Groundwater Management and Optimization

The best and optimum management must not render any harm on the well or the surrounding aquifer. In any groundwater resources management, the pump discharges

must remain slightly below the optimum discharges. In order to avoid additional costs and unnecessary groundwater-level falls, there must not be any inference among the adjacent wells. In case of already existing wells the distances among them are known. These distances are the main guiding information for avoiding any inference between two wells. The optimum solution is neither unique nor constant. The most important question in any groundwater management problem is what amount of discharge is abstractable from each well? For this purpose, the following points must be considered.



**Fig. 4.17** Rough horizontal fracture finite dimensionless fracture aperture factor type curves, **a**  $r = 1.0$ , **b**  $r = 1.5$

- (1) There must not be any material damage in the adjacent well-aquifer joint performance,
- (2) In case of multiple wells, the interference must not be allowed among them.

It is well known that during water abstraction, the groundwater velocity and hydraulic gradient increase toward the main well. If the velocity is more than a critical value, then fine materials may clog the voids near the well and also the filters and such a situation may cause further hydraulic gradient increase leading to undesirable damages near the well vicinity. The groundwater velocity at any point in the aquifer is equal to the Darcy specific discharge (velocity),  $q$ , as,

$$q = Ti \quad (4.11)$$

In this expression,  $T$  is the aquifer transmissivity and  $i$  is the hydraulic gradient. It is possible to state from this equation that transmissivity is equal to the unit specific discharge amount. Darcy velocity,  $v$ , must not be confused with the real velocity. It is calculated as the average velocity that flows from the whole cross-section area including voids and solids. The real velocity can be expressed as the ratio of total discharge to the void area.

$$v_r = \frac{Q}{A_n} = \frac{Av}{A_n} = \frac{v}{A_n/A} = \frac{v}{p} \quad (4.12)$$

Herein,  $p$  indicates the void ratio in the aquifer material. Physically, the groundwater approach velocity to the well is dependent on grain size and hydraulic conductivity. Sichard (1927) found empirically the aquifer well approach velocity,  $v_{aq}$ , in terms of the hydraulic conductivity,  $K$ , as follows.

$$v_{aq} = \frac{\sqrt{K}}{60} \quad (4.13)$$

This expression provides finer numerical results if the hydraulic conductivity is calculated as the ratio of field measurement and type curve matching resultant transmissivity by the saturation thickness. On the other hand, Şen (2015) presented that the optimum discharge,  $Q_{oi}$ , from a well number  $i$  can be calculated according to,

$$Q_{oi} = 2\pi r_{wi} m_i \frac{\sqrt{K}}{60} \quad (i = 1, 2, \dots, n) \quad (4.14)$$

In this expression  $r_{wi}$ ,  $m_i$ , and  $K_i$  are the well radius, aquifer saturation thickness, and aquifer hydraulic conductivity, respectively, and  $n$  is the number of wells. The software for groundwater management according to the above explanations and equations is given below in the MATLAB programming language.

```

function [Swopt,R,Qopt] = OptimumGRoundwaterManagement(I,K,Is)
% This program is written by Zekai Sen on 15 July 2016
% I = Aquifer transmissivity coefficient
% K = Saturation layer thickness
% Is = Iteration number
n=size(I);
r = 0.3; % Well radius 0.3 m
for i=1:n(1)
    for j=1:n(2)
        if K(i,j) == 0
            K(i,j)=1;
        else
        end
        k=(I(i,j)/86400)./K(i,j); % Hydraulic conductivity value is divided by 86400 for
conversion of m2/day to m2/sec.
        if k > 0
            sk=sqrt(k); % Square root of hydraulic conductivity
            % OPTIMIN DRAWDOWN SEARCH BY TRIAL AND ERROR PROCEDURE
            Sw=0.0; % The first value of trial and error
            for l=1:Is
                Sw=Sw+0.1;
                S=((r*sk/60).*log(3000*Sw*sk/r))./k;
                rh=100*abs(S-Sw)/max(S,Sw); % Relative error percentage
                if rh < 5 % Relative error is accepted as %5.
                    Sopt=Sw;
                    Swopt(i,j)=Sopt;
                    R(i,j)=3000*Sopt*sk; % Radius of influence for each well
                    Qopt(i,j)=(2*pi*I(i,j)*Sopt*log(R(i,j)/r))/86400;
                else
                end
            end
        else
            Swopt(i,j)=0;
            R(i,j)=0;
            Qopt(i,j)=0;
        end
    end
end
'OPTIMIN DRAWDOWN (m)'
Swopt
'RADIUS OF INFLUENCE (m)'
R
'OPTIMUM DISCHARGE (m3/s)'
Qopt
end

```

The management of groundwater resources in arid and semiarid regions includes several words and one of the most frequent ones is the “demand” for water. In order to run this MATLAB program, it is necessary to know the transmissivity and hydraulic conductivity values set in the study area. If there is  $n$  number of wells with the transmissivity, and hydraulic conductivity values then there will be  $n$  number of optimum discharges and radius of influence.

As a sample example six wells are assumed in a study area with transmissivity values of  $T = [125 \ 50 \ 80 \ 120 \ 65 \ 210]$

$m^2/day$  and hydraulic conductivities,  $K = [5 \ 12 \ 8 \ 7 \ 13 \ 4] m/day$ . The entrance of these values into the MATLAB program with iteration number 100, the following optimum well drawdown,  $S_{wopt}$ , radius of influence,  $R$ , and optimum well discharge,  $Q_{opt}$  values are obtained.

$$\begin{aligned}
S_{wopt} &= [1.70004 \ 3.0002 \ 7.0002 \ 1.0003 \ 9.0001 \ 2.000] m, \\
R &= [86.752889 \ 583387.142188.741289.004988.7412] m, \text{and} \\
Q_{opt} &= [0.00270 \ 0.00270 \ 0.00280 \ 0.00320 \ 0.0032] m^3/s.
\end{aligned}$$

### 4.5.1 Well Field Optimization

In any management operation, it is significant to arrange the well locations or to calculate the discharges of existing wells in such a way that there are no interferences among them (Meinzer 1920; Walton 1970). Each well has its specific discharge value, which results theoretically in a circle around the well in the aquifer with the radius of influence depending on the aquifer properties and the discharge amount. For no interference case between the two adjacent wells, the summation of these two wells' radius of influences must not be more than the distance between them. The drawdown at any common influence area is equal to the summation of the drawdowns due to each one of them. In a confined aquifer the optimum discharge amount can be calculated in terms of drawdown from well,  $i$ , as,

$$Q_{oi} = \frac{2\pi T_i s_{wi}}{\log\left(\frac{R_i}{r_{wi}}\right)} \quad (i = 1, 2, \dots, n) \quad (4.15)$$

The radius of influence,  $R_i$ , has been given empirically as a function of the drawdown,  $s_{wi}$ , within the pump

well and the aquifer hydraulic conductivity,  $K_i$ , as follows (Şen 2015).

$$R_i = 3000 s_{wi} \sqrt{K_i} \quad (4.16)$$

The substitution of this expression into the previous equation yields the optimum discharge at well  $i$  as,

$$Q_{oi} = \frac{2\pi T_i s_{wi}}{\log\left(\frac{3000 s_{wi} \sqrt{K_i}}{r_{wi}}\right)} \quad (i = 1, 2, \dots, n) \quad (4.17)$$

In order to satisfy simultaneously the two last expressions, one can equate Eqs. (4.15)–(4.17), and finally, the complicated expression for the optimum drawdown in a well can be obtained as,

$$s_{wi} = \frac{r_w \frac{\sqrt{K_i}}{60} \log\left(\frac{3000 s_{wi} \sqrt{K_i}}{r_{wi}}\right)}{K_i} \quad (i = 1, 2, \dots, n) \quad (4.18)$$

Apart from  $s_{wi}$  all the other factors are known in this expression. Hence, the optimum well drawdown can be calculated after a series of trial-and-error approach through the following MATLAB program.

```
function [Sw,R,Qopt] = WellOptimization(T,K,r)
% This program is written by Zekai Sen on 15 JULY 2016
% T = Transmissivity coefficient
% K = Saturation layer (aquifer)thickness
% r = Well radius
% r = 0.3; % Well radius 0.3 m
k=T./K; % HYdraulic conductivity calculation
sk=sqrt(k); % square root of hydraulic conductivity
% OPTIMUM DRAWDOWN CAN BE DETERMINED AFTER TRIAL-ERROR ITERATION
Sw=0; % First value of trial-error approach
for i=1:250
    Sw=Sw+1;
    S=((r*sk/60).*log(3000*Sw*sk/r))./k;
    rh=100*abs(S-Sw)/max(S,Sw); % Relative error percentage
    if rh < Sw
        Swopt=Sw;
    end
end
R=3000*Sw*sk; % HRadius of influence for each well
Qopt=2*pi*T*Sw*log(R/r);
end
```

## References

- Barenblatt, G.I., Zheltov, I.P., and Kochina, I.N., (1960). Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks: Journal of Applied Mathematics, v. 24, 1286–1303.
- Bear, J., (1979). Hydraulics of groundwater: New York, McGrawHill, 567 pp.
- Carslaw, H.S., and Jaeger, J.C., (1947). Conduction of Heat in Solids. Oxford University Press, Oxford, UK.: 510 pp.
- Cooper, H.H. and Jacob, C.E., (1946). A generalized graphical method for evaluating formation constants and summarizing well field history, Am. Geophys. Union Trans., Vol. 27:526–534.
- Darcy, H., (1856). Les Fontaines Publiques de la ville de Dijon. Dalmont, Paris.
- Fetter, C.W., (2001). Applied Hydrogeology. Prentice Hall, Upper Saddle River, NJ, 4th edition. ISBN 0-13-088239-9.
- Freeze, R.A., and Cherry, J.A., (1979). Groundwater: Englewood Cliffs, NJ., Prentice-Hall, 604 p.
- Freeze, R.A., and Cherry, J.A., (1984). Groundwater. Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 604 p.
- Hantush, M.S. (1956). Analysis of data from pumping tests in leaky aquifers. Traps Am. Geophys. Union 37 702–714.
- Kazemi, H., (1969). “A Pressure Transient analysis of Naturally Fractured Reservoirs”, Trans., AIME, 256, 451–461.
- Meinzer, O.E., (1920). Quantitative methods of estimating groundwater supplies. Bull. Geological Society of America 31: 329–338.
- Papadopoulos, I.S., and Cooper, H.H., (1967). Drawdown in a well of large diameter, Water Resources Research, Vol. 3(1): 241–244.
- Sichard, W., (1927). Das Fassungsvermögen von bohrbrunnen und seine bedeutung für grossere absenkiefen. Dissertation, Technische Hochschule, Berlin.
- Snow, D.T., (1965). A parallel plate model of fractured permeable media. PhD Thesis Univ. of Calif., Berkeley, USA.
- Sen, Z., (1985). Volumetric approach to type curves in leaky aquifers.. J. Hydraul. Div. ASCE, 115 (2), 193–209.
- Sen, Z., (1986a). Determination of aquifer parameters by the slope-matching method. Ground Water, Vol. 24(2):217–223.
- Sen, Z., (1986b). Aquifer test analysis in fractured rocks with linear flow pattern. Ground Water 24(1), 72–78.
- Sen, Z., (1995) Applied Hydrogeology for Scientists and Engineers. Taylor and Francis Group, RCR Publishers, Baco Raton, 496 pp.
- Sen, Z., (2015). Practical and Applied Hydrogeology. Elsevier Publishing Co., 406 pp.
- Theis, C. V., (1935). The relation between the lowering of the piezometric surface and the rate and 556 duration of discharge of a well using groundwater storage. Trans. Amer. Geophys. Union, 557 16(1):519–524.
- Walton, W.C., (1970). Groundwater resources evaluation. McGraw-Hill Book Co., New York.
- Warren, J.E., and Root, P.J., (1963). The Behavior of Naturally Fractured Reservoirs, Society of Petroleum Journal, 245–255.



# Hydrochemistry

# 5

## 5.1 General

In the previous chapter, there were methodologies and related software about the quantity of groundwater resources. In practical applications, water quality is another very significant dimension for the planning and joint water resources management. For this purpose, major anion and cation analyses provide basic quality classification information in addition to few composite variable measurements such as the electric conductivity (EC), total dissolved solids (TDS), and sodium adsorption ratio (SAR). Any chemical analysis and related data processing affairs fall within the water quality domain, which is also referred to generally as hydrochemistry. It is the science that is concerned with the natural waters' chemical compositions and provides information about the governing laws explaining changes in the composition. All these activities are consequences of the chemical, physical, and biological processes that take place in the water environment whether in the hydrosphere or within the lithosphere. Essentially, water quality in supply, distribution, and irrigation in the agricultural activities is very important, because hydrochemistry provides verbal knowledge about water quality and numerical data of the water chemical composition. On the other hand, the study of the chemical composition of water becomes particularly important when combatting the pollution of groundwater aquifers against waste waters.

The main purpose of this section is first to provide some basic information about the water quality indicators and then present relevant processing methodologies with the support of MATLAB software programs.

## 5.2 Ion Concentration

In general, major ions in water samples are analyzed in the laboratories and the numerical values for each ion (cations: calcium, Ca; magnesium, Mg; sodium, Na; potassium, K; and anions: chloride, Cl; carbonate,  $\text{CO}_3$ ; bicarbonate,  $\text{H}_2\text{CO}_3$ ; sulfate,  $\text{SO}_4$ ) are expressed in milligram per liter (mg/l); for some rare elements ng/l (nanograms/liter) unit is in use. By conversion definition, 1 mg is 0.001 g and volumetrically 1 l of water is almost equal to 1000 g, and mg/l is also expressible in terms of equivalent parts per million (ppm), and similarly mg/l is equivalent to parts per billion (ppb). Water density is 1 kg/l under standard temperature, pressure, and pure content conditions. In nature groundwater includes various ions (chemical and bacteriological), and therefore, salinity exists to some extent. Due to this non-pure composition of the groundwater, it is possible to say that groundwater in nature has different densities according to salinity, and therefore, an accurate conversion of mg/l to ppm is not possible.

### 5.2.1 Ion Conversion Units and Balance

The most important property of chemical constituents in any water sample is that there should be chemical equilibrium (balance) between total anions and cations, when they are expressed in milli-equivalents per liter (meq/l). This unit is like molality without considering ion exchange. Most often in the laboratory reports ion concentrations are given in ppm, but for the chemical balance they must be converted to

meq/l, because in nature as a result of chemical reactions these ions are generated and changed according to equivalent weights in their environments. Hence, meq/l is a convenient measure for the chemical balance calculations. In the following, the conversion procedures are explained shortly.

The following expression helps to convert ion concentrations from ppm to meq/l. First by definition, ppm is given as,

$$\text{ppm} = \frac{\text{grams of solid}}{10^6 \text{ gram of solution}} \quad (5.1)$$

Equivalent per million is defined as follows.

$$\text{epm} = \frac{\text{ppm}}{10^6 \text{ equivalent weight}} \quad (5.2)$$

On the other hand, the equivalent weight depends on the charge number that is expressible as,

$$\text{equivalent weight} = \frac{\text{atomic weight}}{\text{charge number of the ion}} \quad (5.3)$$

In practical applications, after the necessary conversions, according to Eqs. (5.1)–(5.3), the following two items provide useful information.

- (1) It is not possible to obtain an absolute equivalence between the total anions and cations in meq/l, and therefore, in practical works, it is necessary to accept a certain amount of error, which is frequently adapted at the maximum  $\pm 10\%$ , but preferably  $\pm 5\%$ . In cases of more than these threshold percentage errors the sample must be analyzed again, because there may be some missing constituents or reading and writing errors,
- (2) The total dissolved solids (TDS) value of the same water sample is equal to the summation of anions and cations again in meq/l.

After all what has been said above, the chemical charge balance can be checked reliably by the following ionic charge balance percentage error,  $E$ .

$$E \approx 100 \times \frac{\sum_{i=1}^n (C_i - A_i)}{\sum_{i=1}^n (C_i + A_i)} \quad (5.4)$$

In this expression, cation,  $C_i$ , and anion,  $A_i$ , concentrations are all in meq/l, and various specifications of the major ions and cations are given in Table 5.1.

In the following MATLAB program is given for numerical calculation of major ion and cation balance.

**Table 5.1** Ionic constituent equivalent weights

Ion	Atomic weight (g/mol)	Charge	Equivalent weight
<i>Anions</i>			
Calcium ( $\text{Ca}^{2+}$ )	40.08	+2	20.04
Magnesium ( $\text{Mg}^2$ )	24.32	+2	12.16
Sodium ( $\text{Na}^+$ )	23.00	+1	23.00
Potassium ( $\text{K}^+$ )	39.10	+1	39.10
<i>Cations</i>			
Carbonate ( $\text{CO}^{2+}$ )	60.01	-2	30.00
Bicarbonate ( $\text{HCO}_3^-$ )	61.01	-1	61.01
Sulfate ( $\text{SO}_4^{2-}$ )	96.06	-2	48.03
Chloride ( $\text{Cl}^-$ )	35.46	-1	35.46

```

function [IonsMeq,RE,WWSN] = ChemicalBalance(Ions)
% This program converts ions from ppm to meq/l and then finds percentage
% relative error for each water sample
% Ions is the set of water sample ppm results. It is Nx9 matrix. The first
% column is for sample number, the next four columns are for major cations
% (Ca, Mg, Na and K) and the next last four % columns are for anions
% (Cl, CO3, H2CO3 and SO4): NOTE: Water sample ionic concentrations are
% given in parts per million (ppm)
% N is the number of water samples
% RE is N vector values that includes the relative error for each water
% sample
% WWSN is the water sample number that is not in balance at the 5% relative
% error level
CEW=[20.04 12.16 23.00 39.10]; % Cations (Ca, Mg, Na, K)equivalent weights
AEW=[35.46 30.00 61.01 48.03]; % Anions (Cl, CO3, H2CO3, SO4)equivalent weights
N=length(Ions(:,1)); %Number of water samples
n=Ions(:,1); % Water sample number
% Calculate the miliequivalent per litre values of cations
for i=1:N
    CaMeq(i)=Ions(i,2)/CEW(1); % Ca meq/l
    MgMeq(i)=Ions(i,3)/CEW(2); % Mg meq/l
    NaMeq(i)=Ions(i,4)/CEW(3); % Na meq/l
    KMeq(i)=Ions(i,5)/CEW(4); % K meq/l
    Csum(i)=CaMeq(i)+MgMeq(i)+NaMeq(i)+KMeq(i); % Cation sums in meq/l
end
% Calculate the miliequivalent per litre values of anions
for i=1:N
    ClMeq(i)=Ions(i,6)/AEW(1); % Cl meq/l
    CO3Meq(i)=Ions(i,7)/AEW(2); % CO3 meq/l
    H2CO3Meq(i)=Ions(i,8)/AEW(3); % H2CO3 meq/l
    SO4Meq(i)=Ions(i,9)/AEW(4); % SO4 meq/l
    Asum(i)=ClMeq(i)+CO3Meq(i)+H2CO3Meq(i)+SO4Meq(i); % Anion sums in meq
end
% Arrange the water samples in meq/l matrix
for i=1:N
    IonsMeq(i,1:9)=[n(i) CaMeq(i) MgMeq(i) NaMeq(i) KMeq(i) ...
                    ClMeq(i) CO3Meq(i) H2CO3Meq(i) SO4Meq(i)];
end
% Calculate percentage relative error for each water sample
k=0;
for i=1:N
    RE(i)=100*abs(Csum(i)-Asum(i))/(Csum(i)+Asum(i));
    if RE(i) >= 10
        k=k+1;
        WWSN(k)=n(k); % Wrong Water Sample Number
    else
    end
end
end

```

The application of this MATLAB program to the ion concentrations in Table 5.2 leads to the following results in Table 5.3.

### 5.3 Numerical Indicators

As for the composite water quality indicators, there are several of them that are explained in the following sub-sections.

#### 5.3.1 Electrical Conductivity (EC)

The more the ion concentration the stronger is the electrical conveyance of a water sample, and this is referred to as the electric conductivity (EC). This is the simplest way of assessing the total dissolved solid content of water, which shows the resistance of water to convey the electric current. Hence, EC is equivalent to the reciprocal of resistance, which increases with water salinity. Higher EC readings imply inorganic dissolved solids such as  $\text{Cl}^-$ ,  $\text{SO}_4^-$ ,  $\text{Na}^+$ ,

**Table 5.2** Ion concentrations in ppm

Sample No.	Cations				Anions			
	Ca <sup>2+</sup>	Mg <sup>2+</sup>	Na <sup>+</sup>	K <sup>+</sup>	Cl <sup>-</sup>	CO <sub>3</sub>	H <sub>2</sub> CO <sub>3</sub> <sup>-</sup>	SO <sub>4</sub> <sup>2-</sup>
1	38	19	20	26	26	32	217	8
2	73	25	28	8	40	24	350	16
3	80	35	45	8	46	40	450	20
4	76	26	25	4	40	35	380	16
5	68	26	48	8	56	54	360	16
6	44	22	20	2.6	26	29	250	8
7	70	26	25	4	40	50	390	16
8	105	60	82	10	56	48	360	16
9	22	12	20	2.3	15	25	150	5
10	40	20	27	2.6	26	29	240	10

**Table 5.3** Ion concentrations in meq/l

Sample No.	Cations				Anions				Relative error (%)
	Ca <sup>2+</sup>	Mg <sup>2+</sup>	Na <sup>+</sup>	K <sup>+</sup>	Cl <sup>-</sup>	CO <sub>3</sub>	H <sub>2</sub> CO <sub>3</sub> <sup>-</sup>	SO <sub>4</sub> <sup>2-</sup>	
1	1.8962	1.5625	0.8696	0.665	0.7332	1.0667	3.5568	0.1666	5.03
2	3.6427	2.0559	1.2174	0.2046	1.128	0.8	5.7368	0.3331	5.80
3	3.992	2.8783	1.9565	0.2046	1.2972	1.3333	7.3758	0.4164	7.15
4	3.7924	2.1382	1.087	0.1023	1.128	1.1667	6.2285	0.3331	10.87
5	3.3932	2.1382	2.087	0.2046	1.5792	1.8	5.9007	0.3331	10.26
6	2.1956	1.8092	0.8696	0.0665	0.7332	0.9667	4.0977	0.1666	9.38
7	3.493	2.1382	1.087	0.1023	1.128	1.6667	6.3924	0.3331	16.52
8	5.2395	4.9342	3.5652	0.2558	1.5792	1.6	5.9007	0.3331	19.57
9	1.0978	0.9868	0.8696	0.0588	0.423	0.8333	2.4586	0.1041	11.79
10	1.996	1.6447	1.1739	0.0665	0.7332	0.9667	3.9338	0.2082	8.95

and Ca<sup>2+</sup>. The measurement unit of EC is micro-mhos per cm, where mhos can be read reciprocally (from the right to the left) as ohms. The basic unit of conductivity is the mho or Siemens. After all these information, the specific EC is defined as conductance through one milliliter of water at a standard temperature, 25 °C; an increase of 1 °C causes increases in conductance by approximately 2%. As the temperature increases water conductivity also increases. For comparison purpose among different water samples the conductivity is reported at 25 °C.

The EC is the simplest measure of the water quality classification as indicated in Table 5.4 (Wilcox 1955).

**Table 5.4** EC classification

EC range	Quality classification
<250	Excellent
250–750	Good
750–2000	Permissible
2000–3000	Doubtful
>3000	Unsuitable

### 5.3.2 Total Dissolved Solids (TDS)

This measure is dependent on all solid materials solution in water without suspended sediment colloids and dissolved gases. Its determination in the laboratory is explained by Sen (2016). In general, TDS is related to EC through a linear expression as follows.

$$\text{TDS} = \alpha \text{EC} \quad (5.5)$$

Here, the proportionality coefficient,  $\alpha$ , assumes values between zero and one, but in most of the practical applications, it is close to 0.6. Equation (5.5) implies that the coefficient indicates TDS value per EC. TDS water categorization values are given in Table 5.5.

### 5.3.3 Total Hardness

This is another composite water quality parameter, which is helpful for the classification of domestic and industrial water according to Table 5.6.

**Table 5.5** TDS (mg/l, ppm, g/m<sup>3</sup>)

TDS	Category
<1000	Fresh
1000–10,000	Brackish
10,000–100,000	Saline
>100,000	Brine

**Table 5.6** Hardness classification

TH	Classifications
<75	Soft
75–150	Moderately hard
150–300	Hard
>300	Very hard

**Table 5.7** Irrigation water SAR classification

SAR	Irrigation water class
<10	Very good
10–18	Good
18–26	Medium
>26	Bad

It is expressed in terms of different chemical quantities among which the most effective ones are the  $\text{Ca}^{2+}$  and  $\text{Mg}^{2+}$  ions.

$$\text{TH} = \text{Ca}^{2+} \times \frac{\text{CaCO}_3}{\text{Ca}^{2+}} + \text{Mg}^{2+} \times \frac{\text{CaCO}_3}{\text{Mg}^{2+}} \quad (5.6)$$

### 5.3.4 Sodium Adsorption Ratio (SAR)

This is a special indicator to measure the groundwater convenience possibility for irrigation purposes, which is mainly for determination of defective effect of sodium, and therefore, named as the sodium adsorption ratio. Extra sodium content in groundwaters clogs the soil voids leading to impermeability of water and air through soil and, hence, causes increase in the soil pH. It also indicates the possible solid layer occurrence on the soil surface. It is related to  $\text{Na}^+$ ,  $\text{Ca}^{2+}$ , and  $\text{Mg}^{2+}$  ion concentrations as,

$$\text{SAR} = \frac{\text{Na}^+}{\sqrt{\text{Ca}^{2+} + \text{Mg}^{2+}}} \quad (5.7)$$

SAR is used for irrigation water classification according to Table 5.7, which indicates that as it increases irrigation water usage for agricultural purposes decreases (Sen 2016).

## 5.4 Graphical Representations

Compilation of major ions and composite variables provides a hydrochemical data basis for treatment and collective interpretation about water quality. For this purpose, a graphical representation of available data exposes visual inspection leading to quantitative classifications and inferences. Various researchers have proposed different graphical representations, each with different data processing methodology. For most of the graphical representations the ionic concentrations are expressed in meq/l.

### 5.4.1 Stiff Diagram

This simple categorization is proposed by Stiff (1951) depending on the ion concentrations reflection in a pattern known as Stiff diagrams in the literature. It is based on four parallel horizontal lines graduated numerically on the amounts of anions and cations as in Fig. 5.1. Depending on the ion concentration values an irregular polynomic appears with cation (anion) concentrations on the left (right)-hand side apices.

The following MATLAB program provides opportunity to plot Stiff diagram for any given set of ion concentrations.

```

function StiffDiagram(IonsMeq)
% This program plots Stiff diagram (polygon) for each water sample
% IonsMeq is the set of water sample ppm results. It is Nx9 matrix. The first
% column is for sample number, the next four columns are for major cations
% (Ca, Mg, Na and K) and the next last four % columns are for anions
% (Cl, CO3, H2CO3 and SO4): NOTE: Water sample ionic concentrations are
% given in parts per million (ppm)
% N is the number of water samples
% Detailed explanation goes here
% NOTE: Prior to the run of this program, ionic concentrations in ppm must
% be converted to meq/l. For this purpose, IonsMeq must be taken from a
% previous program "ChemicalBalance"
N=length(IonsMeq(:,1)); % Number of water samples
for i=1:N
    figure
    SNaK=IonsMeq(i,4)+IonsMeq(i,5); % this is the summation of sodium and potassium
    line([-SNaK,IonsMeq(i,6)], [5,5])
    text(-SNaK-0.3,5,'Na+K')
    text(IonsMeq(i,6)+0.1,5,'Cl')
    line([-IonsMeq(i,2),IonsMeq(i,8)], [4,4])
    text(-IonsMeq(i,2)-0.2,4,'Ca')
    text(IonsMeq(i,8)+0.1,4,'H2CO3')
    line([-IonsMeq(i,3),IonsMeq(i,9)], [3,3])
    text(-IonsMeq(i,3)-0.2,3,'Mg')
    text(IonsMeq(i,9)+0.2,3,'SO4')
    line([-SNaK,-IonsMeq(i,2)], [5,4])
    line([-IonsMeq(i,2),-IonsMeq(i,3)], [4,3])
    line([IonsMeq(i,6),IonsMeq(i,8)], [5,4])
    line([IonsMeq(i,8),IonsMeq(i,9)], [4,3])
    title(['Sample number =', num2str(i)])
    axis off
end
end

```

See Fig. 5.2.

In the following figure given the IonsMeq: [n = 1.0000 Ca = 0.3000 Mg = 0.9000 Na = 0.6000 K = 1.1000 Cl = 1.0000 CO<sub>3</sub> = 0.9000 H<sub>2</sub>CO<sub>3</sub> = 0.7000 SO<sub>4</sub> = 0.5000] data the Stiff diagram is obtained from the previous MATLAB program.

#### 5.4.2 Circular Diagram

The basis of the graph is a circle with cation (anion) divisions on the upper (lower) part, where the radial axis is proportional to the total milli-equivalents (Fig. 5.3).

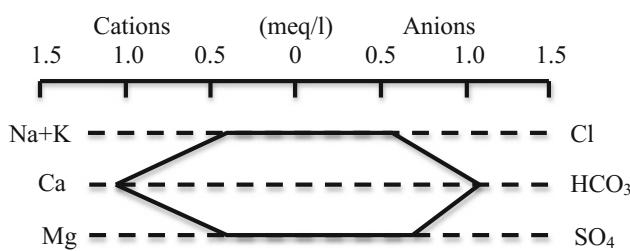


Fig. 5.1 Stiff polygons

The circular diagram is referred to as the pie chart, and its ready software is available in the MATLAB library as follows.

`pie(X)` draws a pie plot of the data in the vector  $X$ . The values in  $X$  are normalized via  $X/\text{SUM}(X)$  to determine the area of each slice of pie. If  $\text{SUM}(X) \leq 1.0$ , the values in  $X$  directly specify the area of the pie slices. Only a partial pie will be drawn if  $\text{SUM}(X) < 1$ .

#### 5.4.3 Schoeller Diagram

This is another graphical representation, where ions are shown on the horizontal axis with their numerical concentrations on the logarithmically graduated vertical axis as in Fig. 5.4 (Schoeller 1967). The benefit from the logarithmic scale is due to its high (low) ion concentration values reduction (enlargement) to get more uniform data scatter.

The following MATLAB program provides the calculation and graphical representation of anions and cations according to the Schoeller diagram.

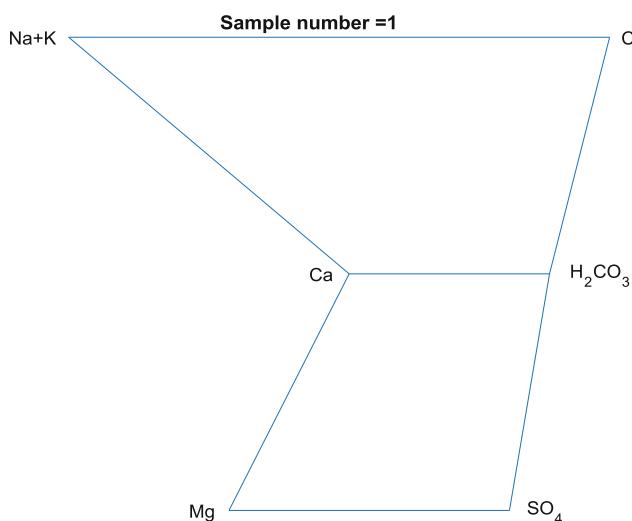
```

function ShoellerDiagram(A,K)
% This program is written on 10 April 2015 by Zekai Sen
% A = Anions in terms of epm
% K = Cations in terms of epm
% AKS = Anion + Cation number
% NOTICE: plot COMMAND IS EQUAL TO THE NUMBER OF SAMPLES
% 1    Na
% 2    K
% 3    Ca
% 4    CO_3
% 5    SO_4
% 6    H_2CO_3
C=[A K];
semilogy(log(C(1,1:end)), 'k')
hold on
semilogy(log(C(2,1:end)), 'b')
semilogy(log(C(3,1:end)), 'r')
semilogy(log(C(4,1:end)), 'k')
semilogy(log(C(5,1:end)), 'g')
legend('Sample 1','Sample 2','Sample 5')
ylabel('Ion concentration (epm)')
xlabel('Ion types')
grid on
box on
title('Shoeller diagram')
end

```

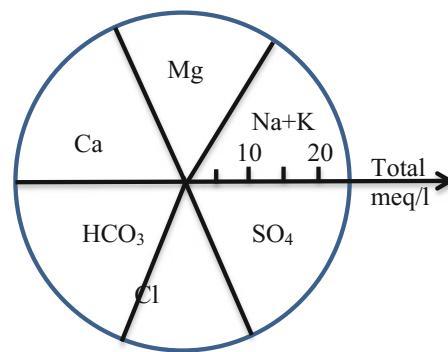
## 5.5 Trilinear (Piper) Diagram

Water chemical constituents are expressed in terms of ions as explained in Sect. 5.2. Various graphical representations for meaningful interpretations are explained in the previous sections. The trilinear diagram is suggested first by Hill (1940) for a more representative description of the water ionic constituents. Later, Piper (1944) refined the trilinear diagram presentation and introduced it in an explicit form. One of the triangles is for anion percentages, which includes



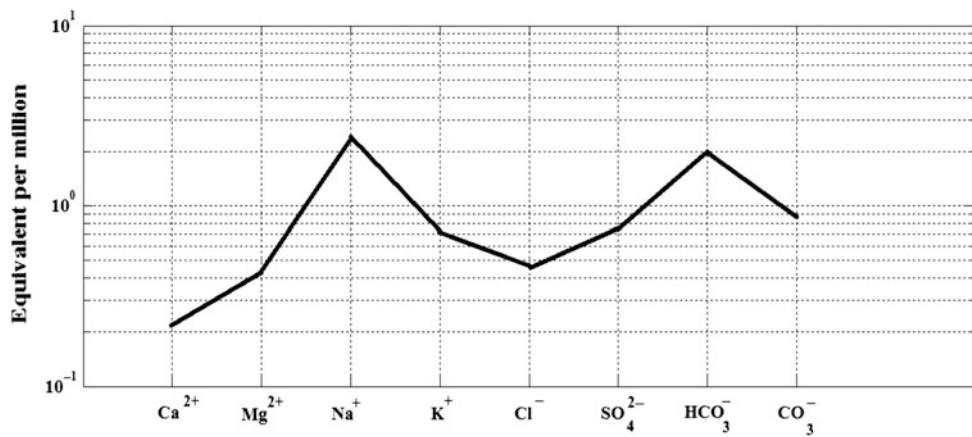
**Fig. 5.2** MATLAB program output Stiff diagrams

three values, and if the number of the anions is more than three some of them must be added together to reduce the number of ionic concentration values down to three. For example, if the anions are Ca<sup>2+</sup>, Mg<sup>2+</sup>, and Na<sup>+</sup> and K<sup>+</sup> then, in general Na<sup>+</sup> + K<sup>+</sup> is the third value. A similar operation is valid for the cations to construct the second triangular diagram. Each side of the triangles is graduated from 0 to 100%, and each triangular area is divided into four sub-regions as in Fig. 5.5. Depending on the percentage graduations in three directions within each triangle, the water sample anion and cation ions are represented by a common point in these triangular areas. Triangular sub-regions inside each anion and cation triangle help to decide dominant and non-dominant cations and anions from their locations in the respective triangles. For example, sub-regions B and H are

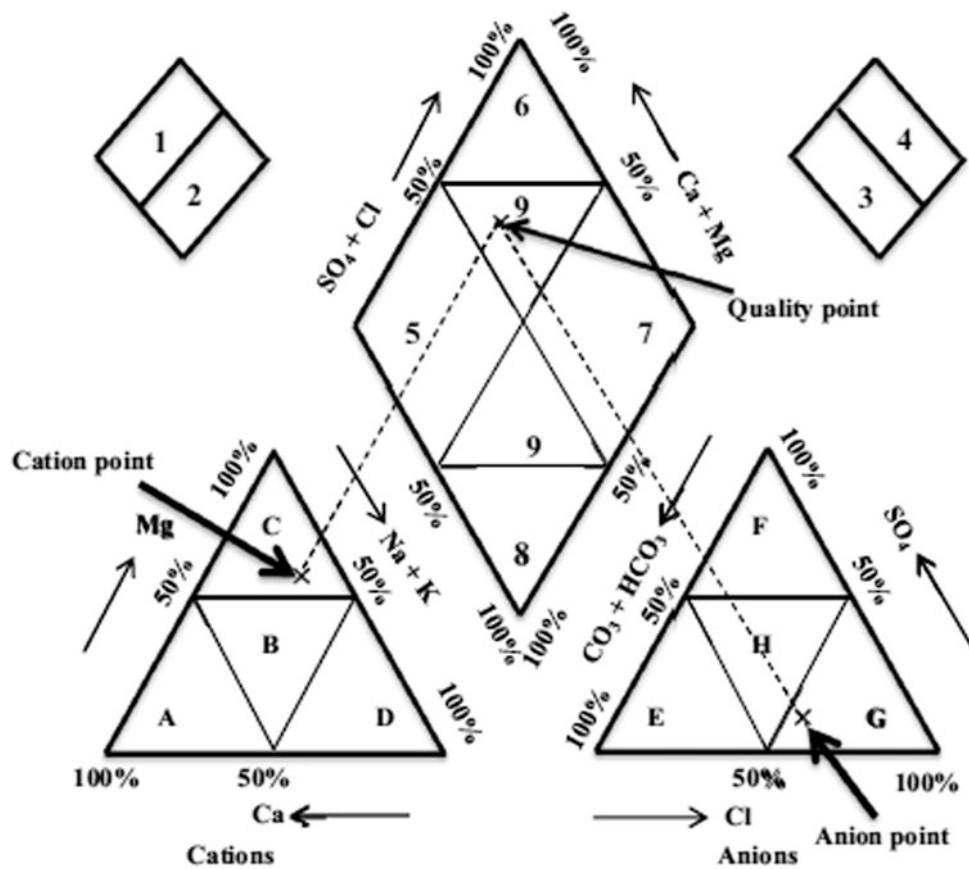


**Fig. 5.3** Circular diagram

**Fig. 5.4** Schoeller water quality diagram



**Fig. 5.5** Trilinear diagram



reserved for non-dominant cation and anion, respectively. Other regions (A, C, D and E, F, G) imply dominant areas for each ion more than 50%. The cation and anion points in the base triangles are then transferred to the rhombohedral area in between the two triangles (see Fig. 5.5).

It is to be noticed from this figure that the area is divided first into four smaller sub-rhombohedral fields as shown at the upper left and right sides. Table 5.8 provides verbal information for each sub-area in the two triangles and in the rhombohedral ionic concentration areas.

```

function ShortVersionTriangularDiamond_PiperGraph(Anion,Cation)
% This program is written on 10 April 2015 by Zekai Sen t
% Anion = Anions in epm (In sequence Ca, Mg, Na and K)
% Cation = Cations in epm (In sequence Cl, SO_4, CO_3, H_2CO_3)
% NOTE: THE ANION AND CATION SEQUENCE CAN BE CHANGES ACCORDING TO THE USER DATA
N=length(Anion(:,1)); % Water sample number
M=length(Anion(1,:)); % Anion (Cation) number
% Convert epm values to percentages
SA=0;
CA=0;
for i=1:M
    SA=SA+Anion(i); % Total anion in epm
    CA=CA+Cation(i); % Total cation in epm
end
for i=1:N
    A(i,1)=100*Anion(i,1)/SA; % Ca percentages
    A(i,2)=100*Anion(i,2)/SA; % Mg percentage
    A(i,3)=100*(Anion(i,3)+Anion(i,4))/SA; % (Na+K) percentage
    C(i,1)=100*Cation(i,1)/CA; % Cl percentages
    C(i,2)=100*Cation(i,2)/CA; % SO_4 percentage
    C(i,3)=100*(Cation(i,3)+Cation(i,4))/CA; % (CO_3+H_2CO_3) percentage
end
% Trigonometrik katsayilar
c60=cos(pi/3);
c30=cos(pi/6);
s60=sin(pi/3);
s30=sin(pi/6);
t60=tan(pi/3);
t30=tan(pi/6);
figure(1)
% ANIONTRIANGULAR DIAGRAM DRAW
line([0 100],[0 0], 'Color', 'k')
text(0,-2,'100%')
text(100,-2,'0%')
text(48,-2,'50%')
text(48,-7,'Ca')
line([0 50],[0 100*c30], 'Color', 'k')
text(40,100*c30,'100%')
text(52,100*c30,'0%')
text(-10,5,'0%')
text(100,5,'100%')
text(18,50*c30,'50%')
text(10,50*c30,'Mg')
text(77,50*c30,'50%')
text(86,50*c30,'Na+K')
line([50 100],[100*c30 0], 'Color', 'k')
axis off
% Triangular net draw
D=[10 20 30 40 50 60 70 80 90];
for i=1:9
    line([D(i)*c60 100-D(i)*c60],[D(i)*c30 D(i)*c30], 'LineStyle', '--', 'Color', 'k')
    line([D(i) 50+D(i)*c60],[0 100*c30-D(i)*c30], 'LineStyle', '--', 'Color', 'g')
    line([D(i) D(i)*c60],[0 D(i)*c30], 'LineStyle', '--', 'Color', 'r')
end
hold on
for i=1:N
    y=A(i,2)*c30;
    x=A(i,1)+A(i,2)*c60;
    plot(x,y, 'k*')
    hold on
end
figure(2)
% CATION TRIANGULAR DIAGRAM DRAW
line([0 100],[0 0], 'Color', 'k')
text(0,-2,'0%')
text(100,-2,'100%')
text(48,-2,'50%')
..... Continued .....
```

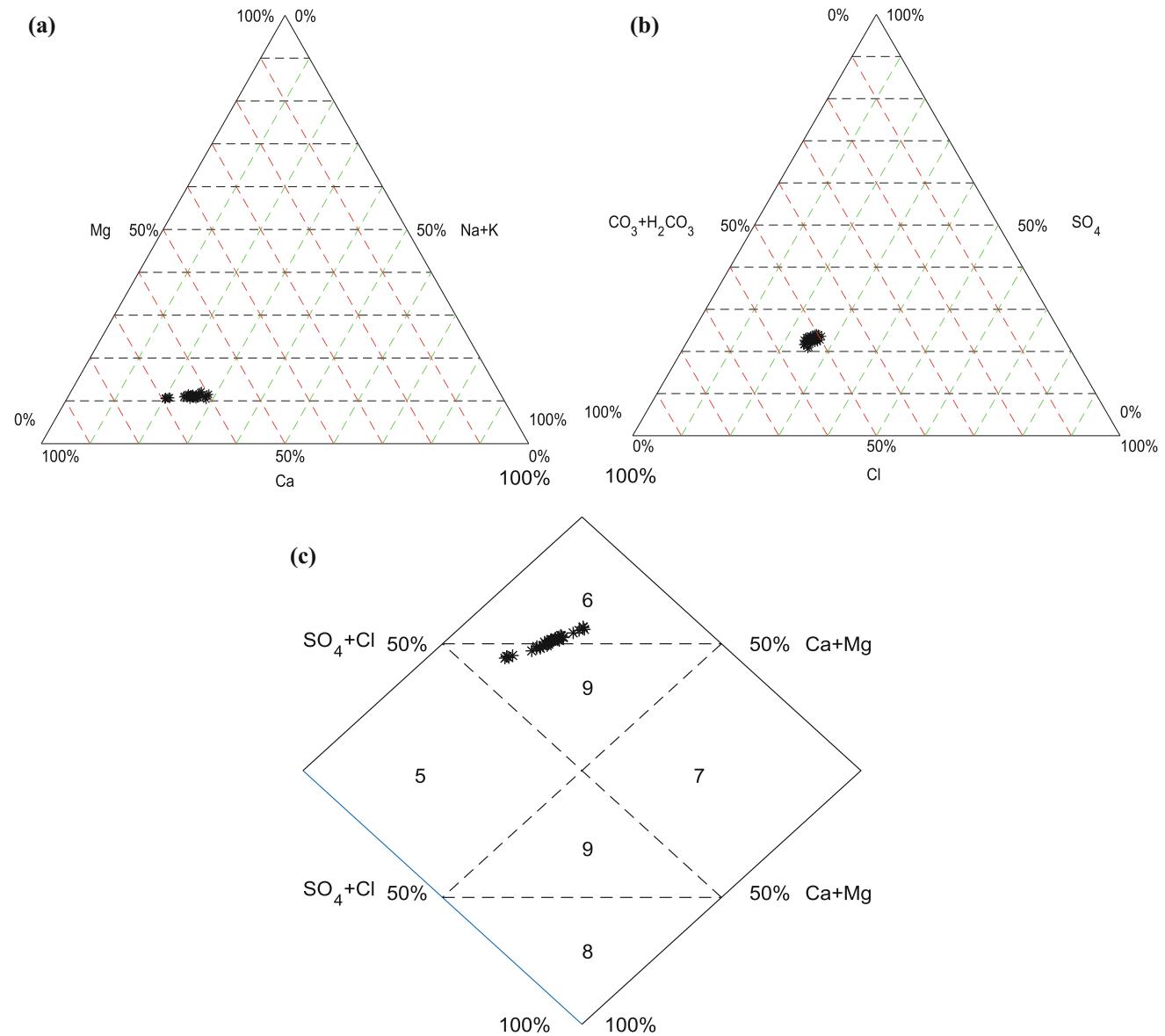
```

text(48,-7,'Cl')
line([0 50],[0 100*c30], 'Color', 'k')
text(40,100*c30, '0%')
text(52,100*c30, '100%')
text(-10,5, '100%')
text(100,5, '0%')
text(18,50*c30, '50%')
text(-5,50*c30, 'CO_3+H_2CO_3')
text(79,50*c30, '50%')
text(90,50*c30, 'SO_4')
line([50 100],[100*c30 0], 'Color', 'k')
axis off
hold on
for i=1:N
    y=C(i,2)*c30;
    x=C(i,1)+C(i,2)*c60;
    plot(x,y,'k*')
    hold on
end
% Triangular net draw
D=[10 20 30 40 50 60 70 80 90];
for i=1:9
    line([D(i)*c60 100-D(i)*c60],[D(i)*c30 D(i)*c30], 'LineStyle', '--', 'Color', 'k')
    line([D(i) 50+D(i)*c60],[0 100*c30-D(i)*c30], 'LineStyle', '--', 'Color', 'g')
    line([D(i) D(i)*c60],[0 D(i)*c30], 'LineStyle', '--', 'Color', 'r')
end
% DIAMOND DRAW
figure(3)
axis off
L=100;
LY=100*c60;
% Diamond periphery draw
line([100*c60 0],[0 100*c30])
text(100*c60-15,0, '100%')
text(100*c60+4,0, '100%')
text(100*c60-15,175, '100%')
text(100*c60+4,175, '100%')
text(100*c60-35,45, '50%')
text(100*c60+30,45, '50%')
text(100*c60-35,130, '50%')
text(100*c60+30,130, '50%')
text(100*c60-50,45, 'SO_4+Cl')
text(100*c60+40,45, 'Ca+Mg')
text(100*c60-50,130, 'SO_4+Cl')
text(100*c60+40,130, 'Ca+Mg')
line([0 100*c60],[100*c30 2*100*c30], 'Color', 'k')
line([100*c60 2*100*c60],[2*100*c30 100*c30], 'Color', 'k')
line([2*100*c60 100*c60],[100*c30 0], 'Color', 'k')
% Diamond net draw
line([L*c60-LY*c60 L*c60+LY*c60],[LY*c30 LY*c30], 'LineStyle', '--', 'Color', 'k')
line([L*c60+LY*c60 L*c60-LY*c60],[LY*c30 L*c30+LY*c30], 'LineStyle', '--', 'Color', 'k')
line([L*c60-LY*c60 L*c60+LY*c60],[L*c30+LY*c30 L*c30+LY*c30], 'LineStyle', '--', 'Color', 'k')
line([L*c60+LY*c60 L*c60-LY*c60],[L*c30+LY*c30 LY*c30], 'LineStyle', '--', 'Color', 'k')
text(50,25, '8')
text(50,60, '9')
text(50,115, '9')
text(50,145, '6')
text(20,85, '5')
text(70,85, '7')
% axis off
% Diamond point mark
hold on
for i=1:N
    y=2*100*c30-(100-C(i,1)-C(i,2)*t30)+A(i,1)-A(i,2)*t30;
    x=y*t30+A(i,1)-A(i,2)*t30-50;
    plot(x,y,'k*')
    hold on
end
end

```

**Table 5.8** Sub-area information in the triangular diagram

	Equilateral triangles	Diamond
Cations	A—Calcium type	1—(Ca + Mg) exceed alkalies (Na + K)
	B—No dominant type	2—Alkalies exceed alkaline earths
	C—Magnesium type	3—Weak acids ( $\text{CO}_3 + \text{HCO}_3$ ) exceed strong acids ( $\text{SO}_4 + \text{Cl}$ )
	D—Sodium and potassium type	4—Strong acids exceeds weak acids
Anions	E—Bicarbonate type	5—Magnesium bicarbonate type
	F—Sulfate type	6—Calcium chloride type
	G—Chloride type	7—Sodium chloride type
	H—No dominant type	8—Sodium bicarbonate type
		9—Mixed type (no cation—anion exceed 50%)



**Fig. 5.6** Trilinear (Piper) graphs: **a** cation percentages, **b** cation percentages, **c** rhomboidal percentages

All what have been explained above about the triangular diagram can be achieved easily through the following MATLAB program.

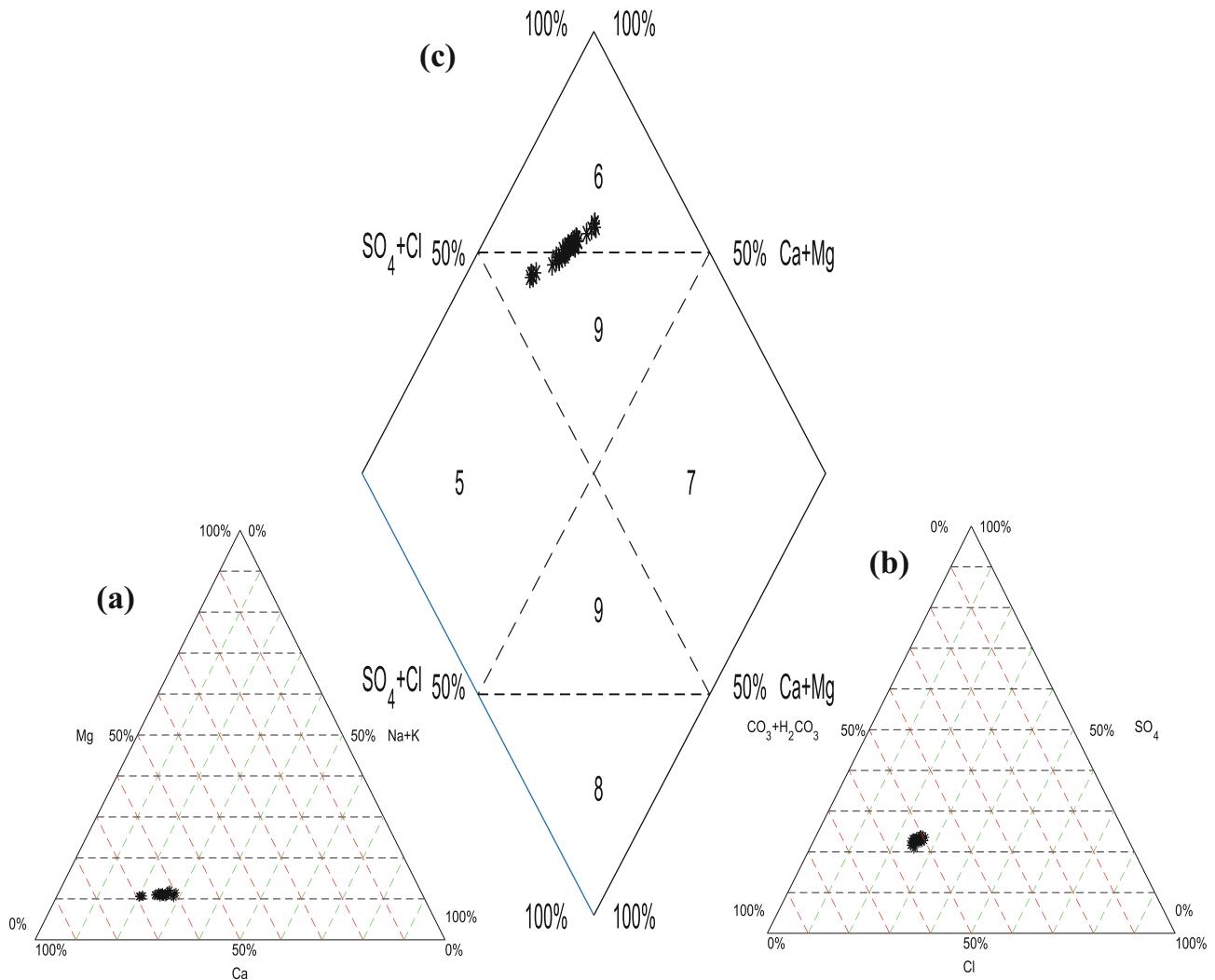
The outcome of this program yields three graphs, namely anion percentage triangle, cation percentage triangle, and rhombohedral graph, which combines the two triangular graphs in a single graph as in Fig. 5.5.

The output of the above MATLAB program yields the two triangular and rhombohedral area descriptions in Fig. 5.6. These graphs can be interpreted under the light of what have been explained above about the triangular diagram.

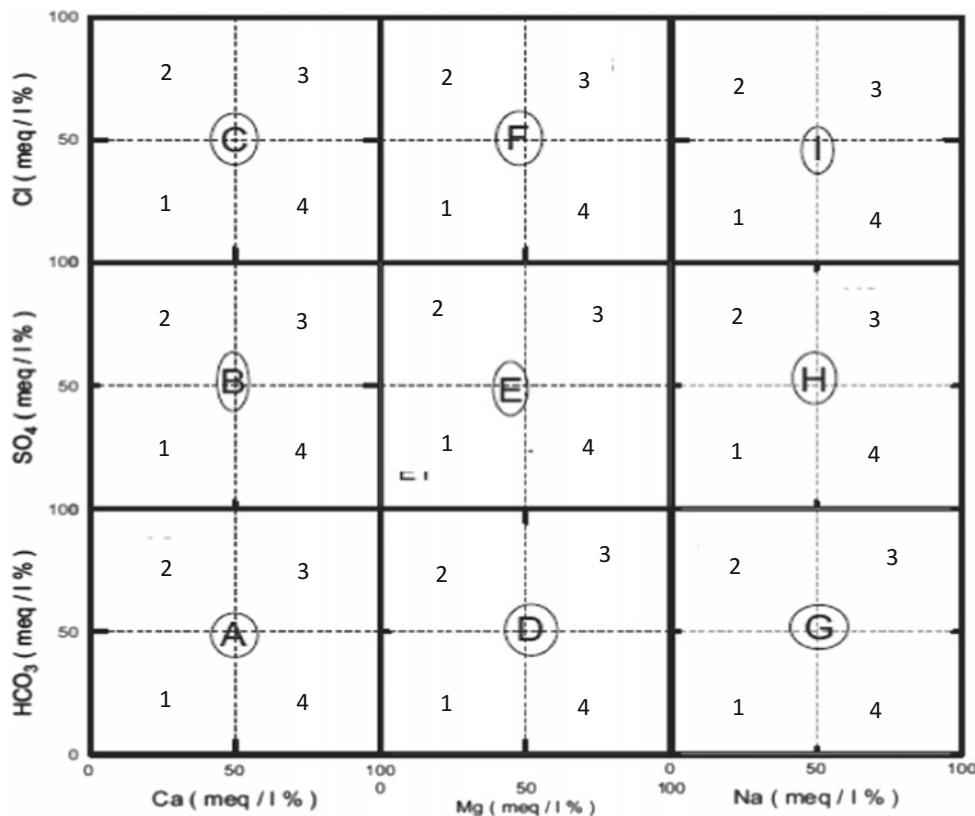
After getting the two triangular and rhombohedral areas as in Fig. 5.6, it is possible to give them the appearance as in Fig. 5.7 by shifting and scaling operations on a Word sheet.

## 5.6 Multi-rectangular Diagram

In the previous section, only six values are considered for anions (3) and cations (3), whereas as the major ions there are at least 4 anions ( $\text{Ca}^{2+}$ ,  $\text{Mg}^{2+}$ ,  $\text{Na}^+$ ,  $\text{K}^+$ ) and four cations ( $\text{Cl}^-$ ,  $\text{CO}_3^{2-}$ ,  $\text{H}_2\text{CO}_3^{2-}$ ,  $\text{SO}_4^{2-}$ ) in practical applications. Triangular diagrams do not provide representation of more than three anions or cations except by reducing their numbers down to three. In case of more than three ions, one can depend on the multi-rectangular diagram (MRD) representation first suggested by Ahmad (1998) and Sen et al. (2003). To overcome the difficulties of classical diagrams for groundwater quality data interpretation, multi-rectangular diagram covers not only the information about the cations and anions separately, but also hydrochemical facies



**Fig. 5.7** Compact trilinear diagrams

**Fig. 5.8**  $4 \times 4$  MRD

classification. In the multi-rectangular diagram (MRD) all major cations (anions) are shown on the horizontal (vertical) axes. A simple MRD is shown in Fig. 5.8 as 3 anions and 3 cations resulting in  $3 \times 3 = 9$  sub-squares. However, if there are 3 anions and 4 cations then  $3 \times 4 = 12$  square MRD can be constructed.

In this figure, the area A with four sub-areas (A1, A2, A3, and A4) represents  $\text{Ca}^{2+}$  and  $\text{H}_2\text{CO}_3^{2-}$  ions associations. In the sub-area A1 (A3), both ions have the least (the most) association with each other. However, in sub-area A2 (A4)  $\text{H}_2\text{CO}_3^{2-}$  anion is more (less) dominant than  $\text{Ca}^{2+}$  cation. The other major squares have similar interpretations with their attached anion and cation values. The MRD provides

information also about the history of groundwater from the present diagram. For example, water samples in sub-square A1 are indicative of fresh groundwaters. The sub-square area A2 shows that ion exchange has occurred in which  $\text{Ca}^{2+}$  has been replaced most probably by  $\text{Na}^+$  from the sediments/rocks through which water has moved. The A3 sub-square gives the indication of calcite dissolution. Finally, water samples A4 are not common but may be possible in coastal areas where more saline seawater intrusion takes place.

In the following, a full-detailed MATLAB program is presented to get the multi-rectangular quality diagram scatter points.

```

function MultiRectangularQualityDiagram(Anion,Cation)
% This program is written on 10 April by Zekai Sen
% A = Anions in epm
% K = Cations in epm
N=length(Anion(:,1)); % Water sample number
M=length(Anion(1,:)); % Anion (Cation) number
% Convert epm values to percentages
SA=0;
CA=0;
for i=1:M
    SA=SA+Anion(i);
    CA=CA+Cation(i);
end
for i=1:N
    % Anion percentages
    A(i,1)=100*Anion(i,1)/SA; % Ca percentages
    A(i,2)=100*Anion(i,2)/SA; % Mg percentage
    A(i,3)=100*Anion(i,3)/SA; % Na percentage
    A(i,4)=100*Anion(i,4)/SA; % K percentage
    % Cation percentages
    C(i,1)=100*Cation(i,1)/CA; % Cl percentages
    C(i,2)=100*Cation(i,2)/CA; % SO_4 percentage
    C(i,3)=100*Cation(i,3)/CA; % CO_3+ percentage
    C(i,4)=100*Cation(i,4)/CA; % H_2CO_3 percentage
end
figure
% axis([0 400 0 400])
box on
% Major lines
line([0 400],[100 100],'Color','k')
line([0 400],[200 200],'Color','k')
line([0 400],[300 300],'Color','k')
line([100 100],[0 400],'Color','k')
line([200 200],[0 400],'Color','k')
line([300 300],[0 400],'Color','k')
% Mid lines
line([0 400],[50 50],'Color','k','LineStyle','--')
line([0 400],[150 150],'Color','k','LineStyle','--')
line([0 400],[250 250],'Color','k','LineStyle','--')
line([0 400],[350 350],'Color','k','LineStyle','--')
line([50 50],[0 400],'Color','k','LineStyle','--')
line([150 150],[0 400],'Color','k','LineStyle','--')
line([250 250],[0 400],'Color','k','LineStyle','--')
line([350 350],[0 400],'Color','k','LineStyle','--')
% Bottom line
text(46,52,'A')
text(21,25,'1')
text(21,73,'2')
text(67,73,'3')
text(67,25,'4')
text(146,52,'E')
text(121,25,'1')
text(121,73,'2')
text(167,73,'3')
text(167,25,'4')
text(246,52,'I')
text(221,25,'1')
text(221,73,'2')
text(267,73,'3')
text(267,25,'4')
text(346,52,'M')
text(321,25,'1')
text(321,73,'2')
text(367,73,'3')
text(367,25,'4')

```

..... Continued .....

```
% Bottom up
text(46,152,'B')
text(21,125,'1')
text(21,173,'2')
text(67,173,'3')
text(67,125,'4')
text(146,152,'F')
text(121,125,'1')
text(121,173,'2')
text(167,173,'3')
text(167,125,'4')
text(246,152,'J')
text(221,125,'1')
text(221,173,'2')
text(267,173,'3')
text(267,125,'4')
text(346,152,'N')
text(321,125,'1')
text(321,173,'2')
text(367,173,'3')
text(367,125,'4')
% Below top
text(46,252,'C')
text(21,225,'1')
text(21,273,'2')
text(67,273,'3')
text(67,225,'4')
text(146,252,'G')
text(121,225,'1')
text(121,273,'2')
text(167,273,'3')
text(167,225,'4')
text(246,252,'K')
text(221,225,'1')
text(221,273,'2')
text(267,273,'3')
text(267,225,'4')
text(346,252,'L')
text(321,225,'1')
text(321,273,'2')
text(367,273,'3')
text(367,225,'4')
% Top layer
text(46,352,'D')
text(21,325,'1')
text(21,373,'2')
text(67,373,'3')
text(67,325,'4')
text(146,352,'G')
text(121,325,'1')
text(121,373,'2')
text(167,373,'3')
text(167,325,'4')
text(246,352,'O')
text(221,325,'1')
text(221,373,'2')
text(267,373,'3')
text(267,325,'4')
text(346,352,'P')
text(321,325,'1')
text(321,373,'2')
text(367,373,'3')
text(367,325,'4')
hold on
```

..... Continued .....

This MATLAB program yields the MRD water quality descriptor in Fig. 5.9 with  $4 \times 4 = 16$  squares each for a specific combination of anions to cations.

```
% Horizontal axis anions percentages
text(40,-36,'Ca%')
text(140,-36,'Mg%')
text(240,-36,'Na%')
text(340,-36,'K%')
% Vertical axis anions percentages
text(-37,50,'Cl%')
text(-47,150,'CO_3%')
text(-58,250,'H_2CO_3%')
text(-47,350,'SO_4%')
% Anion and Cation plots
for i=1:N
    % Ca and anions plot
    plot(A(i,1),C(i,1),'*')
    plot(A(i,1),100+C(i,2),'*')
    plot(A(i,1),200+C(i,3),'*')
    plot(A(i,1),300+C(i,4),'*')
    % Mg and cations plot
    plot(100+A(i,2),C(i,1),'*')
    plot(100+A(i,2),100+C(i,2),'*')
    plot(100+A(i,2),200+C(i,3),'*')
    plot(100+A(i,2),300+C(i,4),'*')
    % Na and cation plots
    plot(200+A(i,1),C(i,1),'*')
    plot(200+A(i,1),100+C(i,2),'*')
    plot(200+A(i,1),200+C(i,3),'*')
    plot(200+A(i,1),300+C(i,4),'*')
    % K and cations plot
    plot(300+A(i,1),C(i,1),'*')
    plot(300+A(i,1),100+C(i,2),'*')
    plot(300+A(i,1),200+C(i,3),'*')
    plot(300+A(i,1),300+C(i,4),'*')
end
xticklabels({'0','%50','%100','%50','%100','%50','%100','%50','%100'})
yticklabels({'0','%50','%100','%50','%100','%50','%100','%50','%100'})
```

Figure 5.9 indicates a sample example for the scatter of cation and anion ions on the multi-rectangular quality diagram.

## 5.7 Standard Ion Index Template

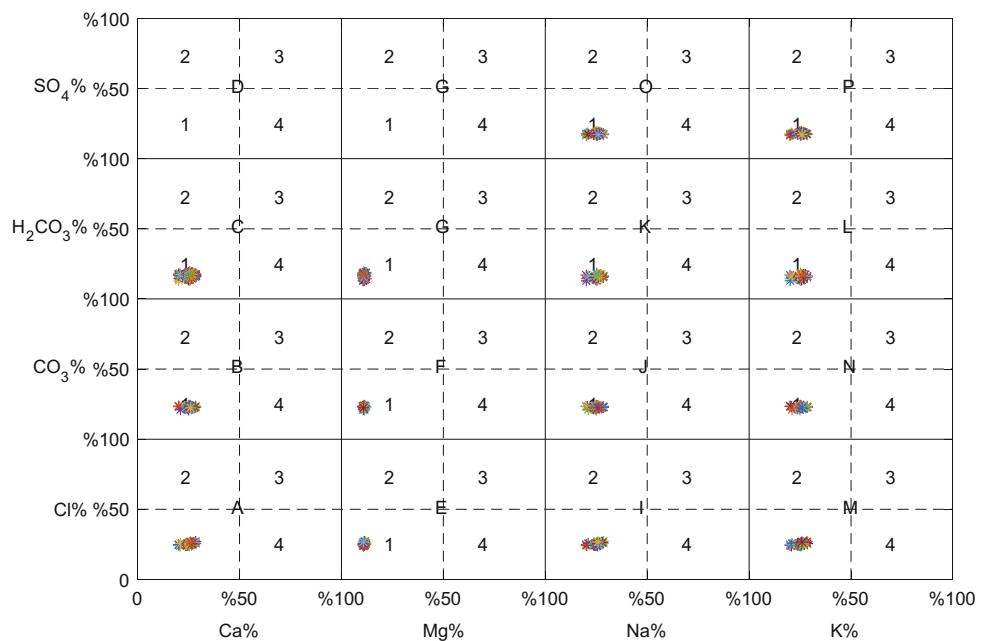
Apart from the previous classical graphical water quality presentations, it is necessary to approach the footprints of water quality by some innovative approaches. Natural water quality does not remain the same with time. In case of water quality data measurements by time, most often their statistical parameters are calculated, and they are compared with

internationally recognized water quality classification charts, and accordingly, a decision is made about the water quality.

Instead of such classical calculation principles, herein, rather simple, innovative, robust, and effective procedure is presented, which is referred to as the “standard ion index (SII)” by taking into consideration all available ion concentrations (Şen 2016). The first step is to standardize the available measurement quality sequence,  $Q_i$ , according to the statistical standardization procedure, which leads to non-dimensional values, with zero arithmetic average and unit standard deviation sequence,  $q_i$ , according to the following expression.

$$q_i = \frac{Q_i - \bar{Q}}{S_Q} \quad (5.8)$$

**Fig. 5.9** Multi-rectangular diagram



where  $\bar{Q}$  and  $S_Q$  are the arithmetic average and the standard deviation of the water quality data. A template is arranged for the comparison of successive standard ion concentrations on the same graph as shown in Fig. 5.10.

This template provides a visual domain to trace the successive ion concentration variation as to whether there are differences between two values and also whether there are clusters of ions. The 1:1 ( $45^{\circ}$ ) straight-line implies that there is not a significant difference between successive samples; however, in case of upward (downward) deviation from the straight-line there is improvement (deterioration) in the

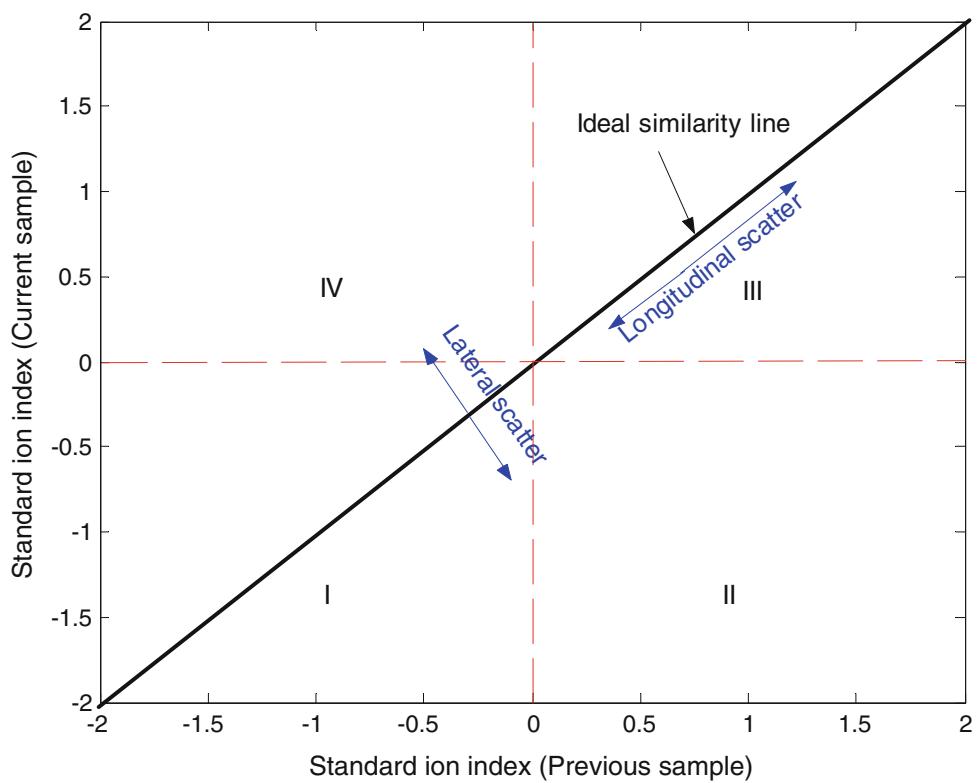
water quality. The closer the points to the 1:1 ( $45^{\circ}$ ) straight-line the more the samples have almost the same water quality. Generally, there are two deviation types between successive sample points, either laterally (getting away from the straight-line) or longitudinally (along the straight-line). The whole variation area in Fig. 5.10 can be divided into several sub-divisions, but herein only four sub-areas are considered as I, II, III, and IV. More detailed information is presented by Sen (2016). The following MATLAB programs help to calculate standard water quality indices.

```

function StandardIonsSuccessivePlots(Ions,Title)
% It standardizes each ion concentration statistically and plots them
% on the same graph
% IonsMeq is a matrix with Nx9 dimension, where the first column is for
% the water sample number sequence, the next four for cations in meq/l
% and the subsequent next four columns are for the anions in meq/l
% av = ion sample average
% st = ion sample standard deviation
% sc = standardized c matrix
% Title = The name pf sample location
N=length(Ions(:,1)); % Number of water samples
for i=1:N
    av(i)= mean(Ions(i,2:9)); % Arithmetic average of each water sample
    st(i)= std(Ions(i,2:9)); % Standard deviation of each water sample
end
% Now calculates and plot sample wise standardized ion values
figure
for i=1:N % Ion concentrations number
for j=2:9 % Water sample number
    sc(i,j) = (Ions(i,j)- av(i))/st(i);
end
end
for i=1:N
    plot(sc(i,2:9));
    hold on
end
grid on
box on
xlabel('1-Ca,          2-Mg,          3-Na,          4-K,          5-Cl,          6-HCO_3,          7-NO_3,          8-
SO_4')
ylabel('Standard ion concentration')
title>Title)
legend('Ca','Mg','Na','K','Cl','CO_3','H_2CO_3''SO_4','Location','NorthWest')
figure
for i=2:N
scatter(sc(i-1,1),sc(i,1),'b*')
end
hold on
for i=2:N
scatter(sc(i-1,2),sc(i,2),'kx')
end
for i=2:N
scatter(sc(i-1,3),sc(i,3),'r+')
end
for i=2:N
scatter(sc(i-1,4),sc(i,4),'g.')
end
for i=2:N
scatter(sc(i-1,5),sc(i,5),'ms')
end
for i=2:N
scatter(sc(i-1,6),sc(i,6),'ys')
end
for i=2:N
scatter(sc(i-1,7),sc(i,7),'gd')
end
for i=2:N
scatter(sc(i-1,8),sc(i,8),'rh')
end
grid on
box on
xlabel('Previous sample')
ylabel('Current sample')
title>Title)
legend('Location','NorthWest','Ca','Mg','Na','K','Cl','HCO_3','NO_3','SO_4')
end

```

**Fig. 5.10** Standard water ionic concentration comparison template



The sample outputs from this MATLAB software are given in Figs. 5.11 and 5.12 for standard ion concentrations and standard successive ion concentrations, respectively. For these two figures, it is necessary to have more than one water sample at a given location. Table 5.9 presents a number of major ion concentrations in ppm.

The template in Fig. 5.9 can be used for time series records of water quality variables or ions. For instance, in Fig. 5.12 an example is provided for ionic concentration scatter diagram on a similar template for groundwater measurements from a single well. It is obvious that 8 major ions on template appear as dominant scatter points in sub-areas I and III.

In this template, the potassium and sodium ions take place low- and high-concentration locations at the lower and higher ends, respectively. On the other hand, longitudinal dispersions are comparatively bigger for the sodium and chloride ions than the others. Finally, comparatively sub-areas II and IV have very small number of scatter points.

## 5.8 Water Quality Index

This section is concentrated on various water quality indices (WQIs), which help to categorize the water samples into different groups. Each one of the WQIs indicates a different feature of water quality. The first and well-known WQI is based on the weighted average of ion concentrations (including anions and cations),  $c_i$ , and it is expressed as,

$$WQI = \frac{\sum_{i=1}^n w_i C_i}{\sum_{i=1}^n w_i} \quad (5.9)$$

where  $w_i$  is the weight of  $i$ th ion and  $C_i$  is the concentration of  $i$ th ion. The weights are taken as water quality guidance of concerned country. For instance, according to SAS (1984) the major ion concentrations should have the permissible limits as  $\text{Ca} < 200 \text{ ppm}$ ;  $\text{Mg} < 150 \text{ ppm}$ ;  $\text{Na} < 200 \text{ ppm}$ ;  $\text{K} < 12 \text{ ppm}$ ,  $\text{Cl} < 600 \text{ ppm}$ ;  $\text{H}_2\text{CO}_3 < 300 \text{ ppm}$ ;  $\text{NO}_3 < 40 \text{ ppm}$ ; and  $\text{SO}_4 < 400 \text{ ppm}$ . Except K and  $\text{H}_2\text{SO}_4$  are not from the

SAS standards. Insertion of these permissible limits into Eq. (5.9) yields explicitly,

$$WQI = \frac{200C_{Ca} + 150C_{Mg} + 200C_{Na} + 12C_K + 600C_{Cl} + 300C_{H_2CO_3} + 40C_{NO_3} + 400C_{SO_4}}{200 + 150 + 200 + 12 + 600 + 300 + 40 + 400}$$

The percentage of contribution of each ion on the WQI can be found by dividing each numerical factor in the numerator by the total of the denominator, which leads to,

$$WQI = 0.15C_{Ca} + 0.08C_{Mg} + 0.15C_{Na} + 0.006C_K + 0.32C_{Cl} + 0.16C_{H_2CO_3} + 0.02C_{NO_3} + 0.21C_{SO_4}$$

sample is treated as representative of a point in eight major ion dimensional ( $Ca$ ,  $Mg$ ,  $Na$ ,  $K$ ,  $Cl$ ,  $H_2CO_3$ ,  $NO_3$ , and  $SO_4$ )

domain. The Euclidian distance between two successive samples, at times  $t$  and  $t - 1$ , can be expressed as,

$$S = \sqrt{(C_{At} - C_{At-1})^2 + (M_{At} - M_{At-1})^2 + (N_{At} - N_{At-1})^2 + (K_{At} - K_{At-1})^2 + (Cl_{At} - Cl_{At-1})^2 + (HCO_{At} - HCO_{At-1})^2 + (NO_{At} - NO_{At-1})^2 + (SO_{At} - SO_{At-1})^2 + }$$

```
function [WQI] = WaterQualityIndex_SAS(Ion)
% This program calculates water quality index for SAS (1984) ion
% concentration procedure
% Ion is the set of water sample ppm results. It is Nx9 matrix. The first
% column is for sample number, the next four columns are for major cations
% (Ca, Mg, Na and K) and the next last four % columns are for anions
% (Cl, CO3, H2CO3 and SO4): NOTE: Water sample ionic concentrations are
% given in parts per million (ppm)
n = length(Ion(:,1)); % Water sample number
m = length(Ion(1,:)); % Ion concentrations are from column 2 to 9
W=[0.15 0.08 0.15 0.006 0.32 0.16 0.02 0.21]; % Ion concentration weights
%according to the sequence of
%Ca, Mg, Na, K, Cl, H_2CO_3,
%NO_3, SO_4
for i=1:n
    WQI(i)=sum(W.*Ion(i,2:m));
end
end
```

The MATLAB program given below helps to calculate this WQI for a given set of groundwater major ion concentrations.

### 5.8.1 Similarity Model

In search for fingerprint behavior for water quality, it is possible to compare successive time-wise chemical data as for their similarity to each other. For this purpose, each

This is the similarity value and its unit may be in ppm or without unit if the standardized ion concentrations are taken into consideration. The following MATLAB program provides systematic calculation of the similarity values among a set of samples.

```

function [simd, row,col,val] = WaterQualitySimilarity (Ions)
% This program finds the Standard similarity values based on the differences,
% Ions is the set of water sample ppm results. It is Nx9 matrix. The first
% column is for sample number, the next four columns are for major cations
% (Ca, Mg, Na and K) and the next last four % columns are for anions
% (Cl, CO3, H2CO3 and SO4): NOTE: Water sample ionic concentrations are
% given in parts per million (ppm)
% n is the number of sample
% m is the number of elements in each sample
% k is the number of similarity values k = n(n-1)/2
% simd is the vector including the difference similarity values
% simc is the vector including the correlation similarity values
% row is the vector of rows of the ascending order similarity,
% col is the vector of columns of ascending order similarity,
% val is the vector of ascending order similarities,
% Find the averages and standard deviations of each sample
n=length(Ions(:,1));
m=length(Ions(1,:));
for i=1:n
    av(i)= mean(Ions(i,2:m));
    st(i)= std(Ions(i,2:m));
end
% Now calculates and plot sample wise standardized ion values
for i=1:n
    sx(i,2:m) = (Ions(i,2:m)- av(i))/st(i);
end
k=0;
for i=1:n-1
    for l=i+1:n
        k=k+1;
        sumd=0.0;
        sumc=0.0;
        for j=2:m
            sumd=sumd+(sx(i,j)-sx(l,j)).^2;      % difference similarity
        end
        simd(i,l)=sumd;                      % difference similarity
    end
end
% Arrance the similarity matrix into a vector named tr
m1=0;
for i=1:n
    for j=i+1:n
        m1=m1+1;
        r(m1)=i;
        c(m1)=j;
        sira(m1)=m1;
        sirac(m1)=m1;
        assimd(m1)=simd(i,j);
        assimdc(m1)=simd(i,j);
    end
end
% Order in ascending order the vector assimd
nn=n*(n-1)/2;
n1=nn-1;
for i=1:nn
    k=0;
    for j=1:n1;
        if assimd(j) > assimd(j+1)
            k=1;
            d=assimd(j);
            assimd(j)= assimd(j+1);
            assimd(j+1)=d;
            e=sira(j);
            sira(j)=sira(j+1);
            sira(j+1)=e;
        end
    end
end

```

..... Continued .....

```

    else
    end
end
if k == 0 end
end
% find the order row, column and value
for i=1:m1
    row(i)=r(sirac(sira(i)));
    col(i)=c(sirac(sira(i)));
    val(i)=assimdc(sirac(sira(i)));
end
end

```

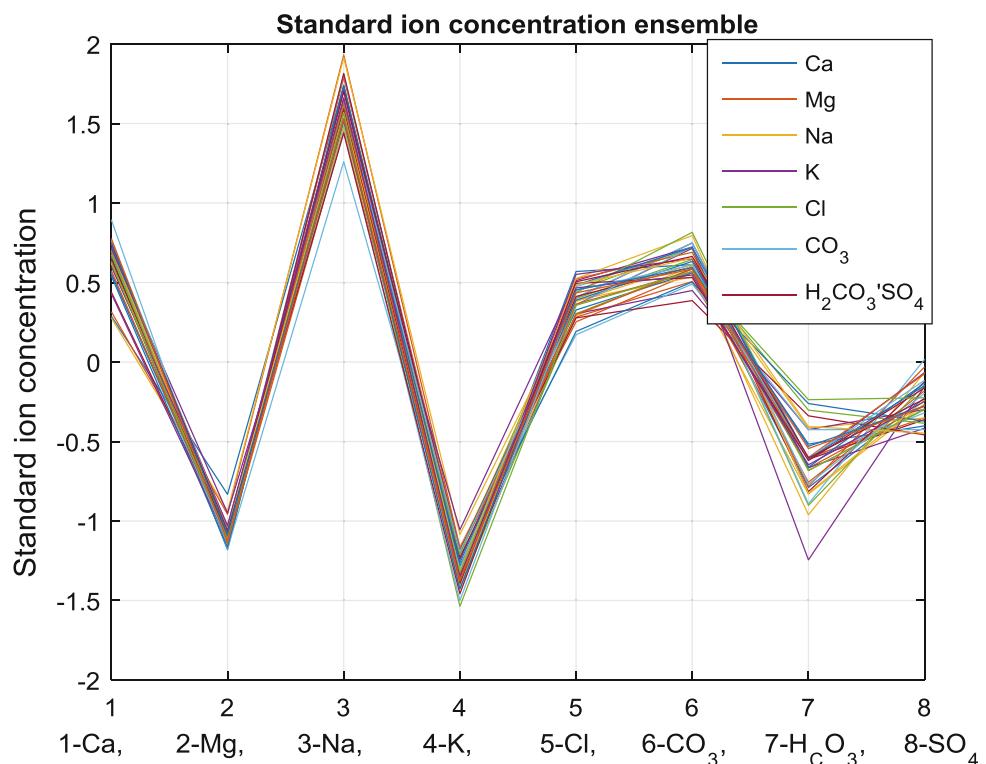
### 5.8.2 Association Matrix

None of the aforementioned methodologies provide sample by sample successive calculation based on each ion. In search for a characteristic signature, it is necessary to visualize all the mutual associations between each pair of the ions. Similar to the chemical association only one of the cations is considered with any one of the anions in an association matrix element calculations. There are several association matrix (AM) representations—the maximum AM (MAAM), minimum AM (MIAM), correlation AM (COAM), and similarity AM (SIAM). For each index some distinctive water quality features are suggested for the first time in this section. In general, an AM representation associates anions with cations. The corresponding template is shown in Fig. 5.13. This figure can be interpreted by

associating each one of the cations in the first column with each of the anions in the first line resulting in a joint value. This means that the type of water samples is determined according to a criterion and included in the AM. It is like the work done about the groundwater chemistry by Sen and Al-Dakheel (1986). For instance, say  $\text{NaH}_2\text{CO}_3$ , it will have an associative value of  $V_{32}$ . In this manner, 16 different types of ion combinations can be viewed easily.

The filling of the values in this template serves the desired goal according to the type of work. For instance, MAAM includes the numbers of maxima association between a cation with an anion. Hence, if there are  $n$  water samples, then the summation of all values in the AM should be equal to  $n$ . For various water quality index calculations the following simple MATLAB program is given.

**Fig. 5.11** Standard ion concentration ensembles



```

function [MAAM, MIAM] = WaterQualityMaximumAssociationMatrix(Ion)
% This program finds maximum association between maximum cations with
% maximum anions
% Ion is the set of water sample ppm results. It is Nx9 matrix. The first
% column is for sample number, the next four columns are for major cations
% (Ca, Mg, Na and K) and the next last four
% columns are for anions (Cl, CO3, H2CO3 and SO4)
% NOTE: Water sample ionic concentrations are
% given in parts per million (ppm)
% MAAM is the maximum association matrix
% MIAM is the minimum association matrix
n=length(Ion(:,1));
m=length(Ion(1,:));
for i=1:4
    for j=1:4
        MAAM(i,j)=0.0;
        MIAM(i,j)=0.0;
    end
end
for k=1:n
    for i=1:4
        for j=1:4
            %Find maximum associate matrix elements
            [A I]=max(Ion(i,2:5));
            [C J]=max(Ion(i,6:9));
            MAAM(I,J)=MAAM(I,J)+1;
            %Find minimum associate matrix elements
            [A I]=min(Ion(i,2:5));
            [C J]=min(Ion(i,6:9));
            MIAM(I,J)=MIAM(I,J)+1;
        end
    end
end

```

### 5.8.3 Fuzzy Classification

Linguistic variables are the most fundamental elements in human knowledge exposition and dissemination. The introduction of these variables gives the opportunity to formulate vague descriptions in natural language in a precise mathematical manner (Sen 2009).

The fuzzy concepts have linguistic basis, and therefore, the foundation elements are words and their derivatives with adjectives. In some publications the fuzzy operations are called as “word computation” (Zadeh 1999, 2001). The transition from the words to numerical values is presented through the membership degrees (MDs) and membership functions (MFs). Classes with non-crisp boundaries are the basis of fuzzy sets. The transition between the classical crisp boundary sets and the fuzzy sets are interchangeable. Fuzzy logic (FL) is a universal principle and concept. An analysis can be based on human fuzzy perceptions under the light of incomplete and vague information and personal experience

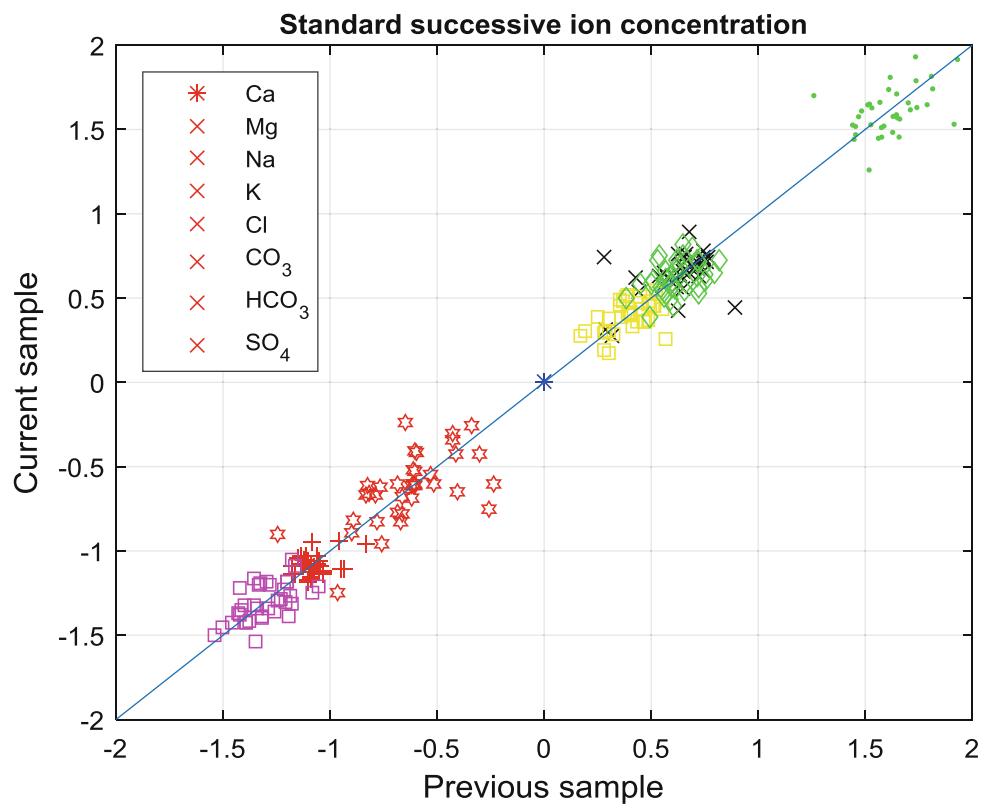
even though it may be subjective. A very detailed explanation of fuzzy logic problem-solving with a set of rules is presented in a textbook by Ross (1995).

## 5.9 Applications

The groundwater hydrochemistry records are used for the implementation of the Kriging methodology so as to obtain triple diagrams that give the common behavior of three variables. Herein, three distinctive but complementary investigations are considered. These are,

- (1) Triple diagrams are constructed directly from major anions and cations, so as to consider three major anions and/or cations common behaviors within the study area. For instance, the triple diagram of the equal TDS concentrations based on Cl and HCO<sub>3</sub> values is presented in Fig. 5.14.

**Fig. 5.12** Successive SII graph  
(Sen 2015)



**Table 5.9** Major ion concentrations from the same well at different dates

Sample No.	Ions are in equivalent per mole (epm)							
	Ca	Mg	Na	K	Cl	$\text{CO}_3$	$\text{HCO}_3$	$\text{SO}_4$
1	3.538	2.09	5.407	1.412	3.901	3.957	2.145	2.873
2	3.713	1.893	6.02	1.596	3.622	4.062	2.307	2.873
3	3.623	1.843	6.02	1.627	3.787	3.999	2	2.832
4	4.207	1.819	5.224	1.637	3.767	4.213	2.387	2.707
5	3.997	1.835	5.22	1.537	3.671	3.98	2.307	2.79
6	4.142	1.819	5.059	1.535	3.716	4.143	2.452	2.811
7	4.187	1.926	5.329	1.535	3.787	4.028	2.468	2.79
8	4.172	1.769	5.677	1.683	3.76	4.079	2.613	2.769
9	4.321	1.761	6.025	1.627	3.792	4.124	2.484	2.852
10	4.251	1.819	5.507	1.601	3.719	4.134	2.79	2.873
11	4.232	1.745	5.685	1.246	3.672	3.881	2.629	2.811
12	4.232	1.884	5.438	1.448	3.716	4.088	2.92	2.811
13	4.371	1.852	6.003	1.652	3.648	4.107	2.79	2.79
14	4.436	1.835	6.16	1.645	3.881	4.042	2.968	2.79
15	4.242	1.728	5.72	1.589	3.604	4.029	2.984	2.832
16	4.227	1.662	5.403	1.581	3.605	3.989	2.178	2.956
17	4.067	1.646	5.237	1.514	3.621	3.938	1.807	3.019
18	4.022	1.695	5.059	1.548	3.741	4.037	1.284	2.811
19	4.062	1.662	5.194	1.519	3.756	4.115	1.936	2.956

(continued)

**Table 5.9** (continued)

Sample No.	Ions are in equivalent per mole (epm)							
	Ca	Mg	Na	K	Cl	CO <sub>3</sub>	HCO <sub>3</sub>	SO <sub>4</sub>
20	4.217	1.687	4.663	1.535	3.701	4.007	2.032	3.144
21	3.583	1.712	5.124	1.599	3.485	3.92	2.049	2.894
22	3.807	1.703	5.185	1.581	3.619	3.92	2.323	2.977
23	3.762	1.564	4.785	1.287	3.625	3.833	2	2.936
24	3.802	1.909	4.763	1.581	3.686	3.994	2.145	2.811
25	3.563	1.794	4.868	1.479	3.701	3.814	2.371	2.998
26	3.827	1.777	4.902	1.486	3.607	4.064	2.258	2.727
27	3.822	1.753	5.146	1.675	3.569	4.017	2.194	2.894
28	4.057	1.835	5.255	1.45	3.859	3.906	2.436	3.019
29	3.772	1.852	4.894	1.473	3.69	3.966	2.42	2.956
30	3.922	1.794	5.281	1.456	3.656	3.914	2.436	3.102
31	3.957	1.893	5.124	1.45	3.791	3.999	2.694	2.644
32	3.972	1.901	5.32	1.491	3.801	3.904	2.387	2.894
33	3.992	1.959	5.081	1.468	3.676	3.924	2.984	2.998
34	4.127	2.008	4.92	1.491	3.76	3.946	2.532	3.102
35	4.127	1.926	4.972	1.445	3.833	4.027	2.484	2.936
36	4.152	1.901	5.12	1.473	3.796	3.967	2.581	2.894
37	4.182	1.802	5.12	1.499	3.755	3.942	2.532	2.863

	Cl	H <sub>2</sub> CO <sub>3</sub>	NO <sub>3</sub>	SO <sub>4</sub>
Ca	V <sub>11</sub>	V <sub>12</sub>	V <sub>13</sub>	V <sub>14</sub>
Mg	V <sub>21</sub>	V <sub>22</sub>	V <sub>23</sub>	V <sub>24</sub>
Na	V <sub>31</sub>	V <sub>32</sub>	V <sub>33</sub>	V <sub>34</sub>
K	V <sub>41</sub>	V <sub>42</sub>	V <sub>43</sub>	V <sub>44</sub>

**Fig. 5.13** The template of associative matrix

It is also helpful to look at the three-dimensional (3D) surface relationship between these chemical constituents as in Fig. 5.15.

The interpretation of the triple diagram map and 3D surface leads to the following logical inferences concerning high TDS concentration rates based on Cl and HCO<sub>3</sub>.

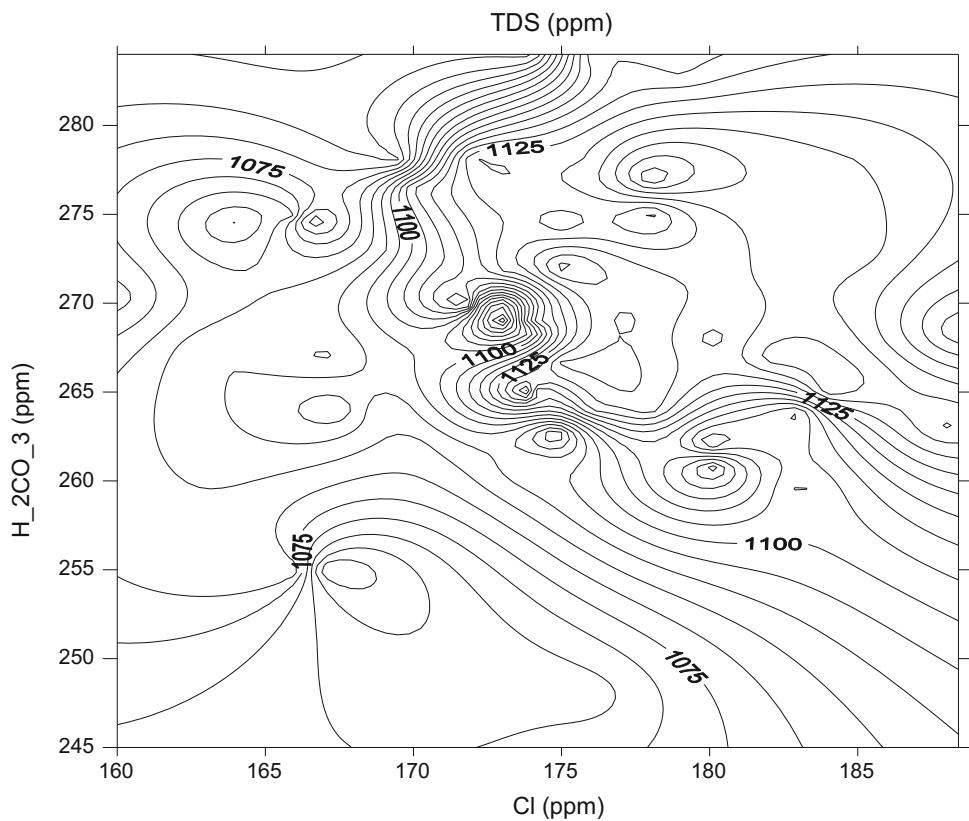
R1 : IF Cl is medium AND HCO<sub>3</sub> is low THEN TDS is high  
OR

R2 : IF Cl is medium AND HCO<sub>3</sub> is medium THEN TDS is very high  
OR

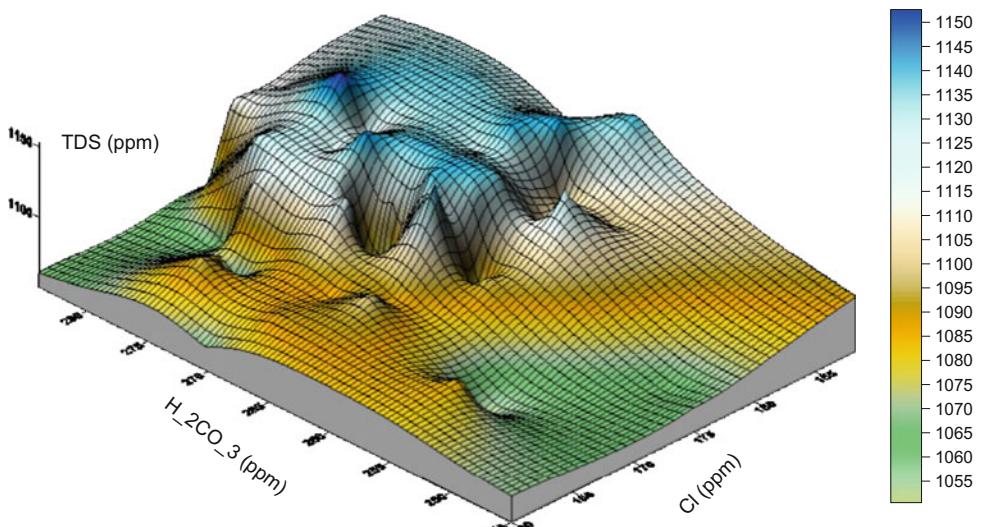
R3 : IF Cl is high AND HCO<sub>3</sub> is low THEN TDS is high.

These logical statements lead hydrogeologists to think about the possibilities of each IF ... THEN rule on the basis of geological subsurface composition of the study area, in addition to the hydrological and hydrogeological features interactions. In this manner, it is possible to obtain clues for reasons of groundwater quality variations. On the other hand, these logical statements provide a common basis for the general variability description of ions within the study area. Such rule bases are prerequisites for fuzzy logic modeling as suggested by Zadeh (1965).

**Fig. 5.14** Equal TDS lines based on Cl and  $\text{HCO}_3$



**Fig. 5.15** Three-dimensional TDS change with Cl and  $\text{HCO}_3$



## References

- Bayındır, Y., (2006). Harran Ovası Serbest Akiferinin Kirlilik Potansiyelinin Belirlenmesi. Harran Üniversitesi, Fen Bilimleri Enstitüsü, Çevre Mühendisliği Ana Bilim Dalı. (in Turkish).
- Davis, S.N., and DeWiest, R.J., (1966). Hydrogeology, New York: Wiley.
- Hem, J.D., (1985). Study and interpretation of the chemical characteristics of natural water. USGS, Water Supply Paper 2254, pp 264.
- Piper, A.M. (1944). A graphical procedure in the geochemical interpretation of water analysis. Am Geophys Union Trans 25:914–928.
- Ross, T. J., (1995). Fuzzy logic with engineering applications. McGraw-Hill.
- Schoeller, H., (1967). Geochemistry of groundwater. An international guide for research and practice” UNESCO, Chapter 15, pp. 1–18.
- Stiff, H.A., Jr., (1951). The interpretation of chemical water analysis by means of patterns: Journal of Petroleum Technology, v. 3. no. 10, p. 15–17.

- Sen, Z., (2016) Applied and Practical Hydrogeology. Elsevier, Amsterdam: 496 pp.
- Sen, Z., (2009). Fuzzy Groundwater Classification Rule Derivation from Quality Maps. Water Quality and Health, Vol. 1: 115.
- Zadeh, L. A., (1999). From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions. IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, 45:105–119.
- Zadeh, L.A., (2001). The problem of deduction in an environment of imprecision, uncertainty, and partial truth. In: Nikravesh M, Azvine B (eds), FLINT 2001, New Directions in Enhancing the Power of the Internet, UC Berkeley Electronics Research Laboratory, Memorandum No. UCB/ERL M01/28.
- Wilcox, L.V., (1955). Classification and use of irrigation waters. USDA, Circular 969, Washington, DC, USA.

# Simulations

## 6.1 General

The natural behavior of earth systems is rather complex and their replicates can be generated by various stochastic models, which take into consideration deterministic and uncertainty components temporally, spatially, or spatio-temporally. As for the temporal variations, there are two types, namely records at regular time intervals (hour, week, and month, annual) that are referred to as time series, which are encountered frequently not only in the earth systems domain but in many other disciplines from electronics, economics, to social sciences. These are modeled by a set of different stochastic processes (Box and Jenkins 1970; Cox and Miller 1965). The second type is records at irregular time interval basis such as earthquake and flood occurrences, and their treatment needs different modeling and simulation procedures, which are frequently probabilistic or statistical methodologies.

Simulation means generation of a set of replicates that are statistically indistinguishable from the original records. It is not possible to obtain exactly the same pattern of the records, but rather very close patterns that fluctuate around the measurement series. Some of the replicates are partially above and below the records such that the statistical parameters are indistinguishably equal to each other. It is possible to generate numerous replicates and as for the question of which one to choose for practical applications among them to represent the future extension of the records (predictions), is the most significant goal to decide. The bunch of replicates is referred to as an ensemble. After the generation of replicates with statistical similarity to the basic measurement sequence, one of the following tasks may serve the purpose.

- (1) The ensemble averages may serve the purpose, and hence, the arithmetic average of each corresponding data to a particular specification is taken according to day, month, or year duration at a specific location,

- (2) There is a possibility to identify a set of extreme values along the time axis or spatially at a set of points, which provide a basis for future extreme event assessment,
- (3) The ensemble members can be treated with a convenient probability distribution function (PDF) for the purpose of depicting the risk levels as 1, 5, or 10%,
- (4) Statistical parameter estimations provide a set of similar values, but in a random fashion, and hence, it is possible to assess the biasness of the generations,
- (5) The overall representative pattern can be decided by taking the weighted average based on each replicate with a certain percentage. This is preferable rather than taking the arithmetic average.

The main purpose of this chapter is to propose various stochastic processes for simulation by means of MATLAB language programs.

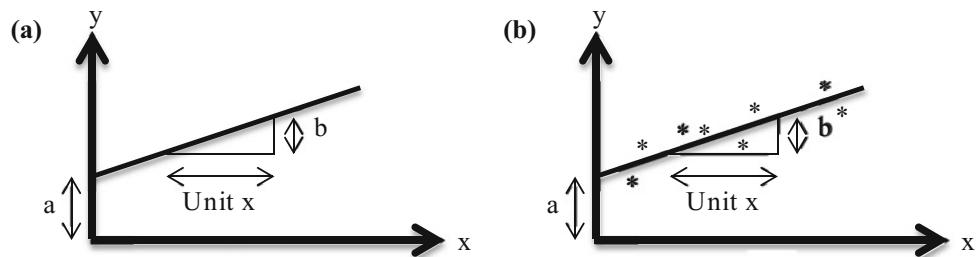
## 6.2 Stochastic Processes

Stochastic processes are the main mathematical probabilistic-statistics tools that help to generate similarity to a given sequence of measurements through effective software. In general, any mathematical expression in the deterministic domain can be converted to the uncertainty domain by adding a random component. The simplest deterministic form is well-known straight-line mathematical linear relationship between an independent,  $x$ , and dependent,  $y$ , variables as follows,

$$y = a + bx \quad (6.1)$$

Here,  $a$  and  $b$  are the model parameters as the intercept and the slope of the straight-line, respectively, and they are shown in Fig. 6.1a.

Conversion of Eq. (6.1) into an uncertain domain is referred to as a stochastic process, and for this purpose, a random term,  $\varepsilon$ , must be added on the right-hand side as (see Fig. 6.1b),

**Fig. 6.1** Linear models

$$y = a + bx + \varepsilon \quad (6.2)$$

In this form, the parameters  $a$  and  $b$  gain different meanings, where  $a$  is the bias and  $b$  is a function of the correlation structure in time series analysis or regression coefficient in classical regression analysis. For the best model, the random numbers must abide with the normal (Gaussian) PDF of zero mean in a bell shape distribution function (Fig. 6.2). The standard deviation of the error term can take any value, but the closer it is to zero, the more valid is the deterministic component for model representation. Theoretically, for complete determinism the standard deviation should also be equal to zero, which is not possible in natural event records. It is better to stick to possible least standard deviation for the best model identification.

It is possible to generate a sequence of  $\varepsilon$  values with these statistical parameters. If the arithmetic average of both sides in Eq. (6.2) is taken, then it reduces to Eq. (6.1). This line must be fitted in such a way that the following conditions are satisfied.

- (1) The vertical (even horizontal or line perpendicular) deviations from the straight-line should have zero arithmetic mean,

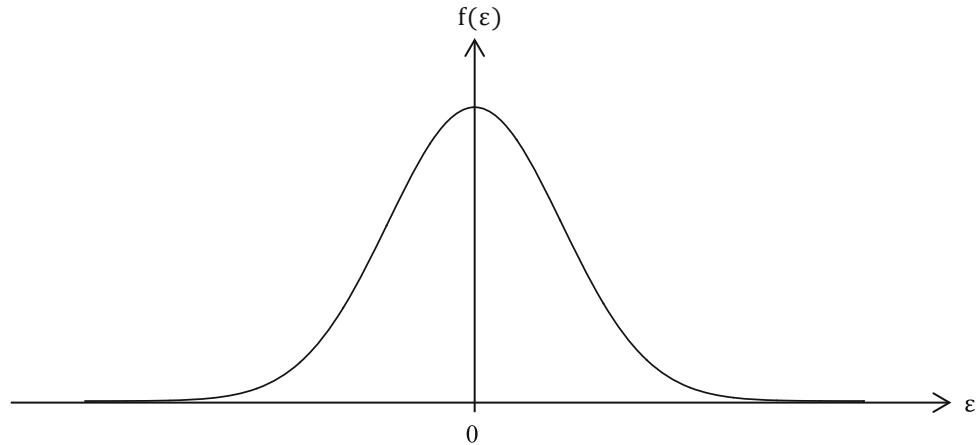
- (2) The arithmetic average of the square deviations cannot be zero, but it must be as small as possible, which is referred to as the least squares technique in the literature.

These are the two basic conditions that should be satisfied procedurally in any simulation or curve fitting methodology. However, to be more precise, the following points should also be satisfied, but they are missed from the care of most researchers.

- (1) The variables ( $x$  and  $y$ ) or at least the residuals should abide with the normal (Gaussian) PDF, and if not then some transformation procedures must be applied to the variables. This is also a requirement for the valid correlation coefficient calculations,
- (2) The residuals should not have any internal (serial) correlation, which must be practically equal to zero,

One can generalize Eq. (6.2) by consideration of any mathematical functional form as  $f(x)$ .

$$y = f(x) + \varepsilon \quad (6.3)$$

**Fig. 6.2** Normal (Gaussian) PDF

Generally, in the literature for simulation studies,  $f(x)$  is considered as a linear function, but it may also be quite complicated mathematical expression like the one in frequent use in the simulation studies as the Fourier (Harmonic) analysis, which has already been explained in Chap. 2.

The most significant structural parameter for any time series apart from the statistical parameters is the serial correlation coefficients of various lags or theoretical autocorrelation functions, which reflect the internal dependence structure of a time series in the form of a function representing the correlation coefficient value on the vertical axis with a succession of lags on the horizontal axis (see Fig. 6.3). By definition theoretically autocorrelation function may take any value between  $-1$  and  $+1$ .

All the aforementioned procedures, mathematical formulations and applications need a set of numerical data concerning the variables, which have in their structures not only random and stochastic components, but also some deterministic parts that are expressible by mathematical functions.

It is important to mention at this stage, the difference of stochasticity from randomness and the determinism. Let us discuss this point from a time series point of view. Any time series,  $T_i$  ( $i = 1, 2, \dots, n$ ), with  $n$  measurement sequence, has generally two components as deterministic,  $D_i$  and uncertainty,  $U_i$  parts.

$$T_i = D_i + U_i \quad (6.4)$$

After the separation of mathematical part (deterministic component) the remaining uncertainty component may still have some remnant or pseudo-determinism, which is called as the stochastic part,  $S_i$ , and when it is filtered from the uncertainty component then the remaining are referred to as residuals that should be explainable only by probability distribution functions (PDFs), because they do not have any

deterministic mathematical structure, so they are completely random,  $\varepsilon_i$ .

$$U_i = S_i + \varepsilon_i \quad (6.5)$$

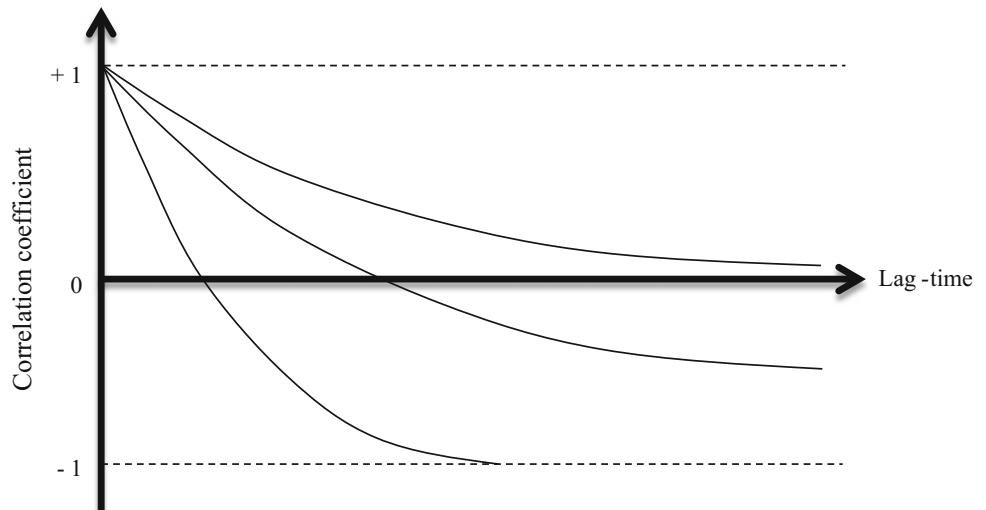
The simulation studies are concerned with the stochastic component only, and after the generation of replicates, one can then rewind the previous steps by adding onto the replicates the deterministic (mathematically expressible) component, which leads to the simulation of the original data or time sequence. There are ready software routines in different libraries (Fortran, MATLAB, and the like) for completely random variable generation, but the remaining task is to identify first mathematical component generally through regression methodology, and then the stochastic component can be modeled by considering one of the procedures in the following sub-sections.

Last but not least, the deterministic part may include seasonality, which can be dealt with harmonic analysis as suggested in Chap. 2 and also trends that are explained in Chap. 10.

### 6.2.1 Independent Process

If any natural or artificial event has completely independent structure without any deterministic (mathematical) and stochastic components, then the only way to treat them scientifically is by means of a convenient PDF. Among some examples are the flood, earthquake, traffic accident, sea surface wave, atmospheric pollution, diffusion, and similar events. Almost all risk calculations in engineering works, insurance and production managements are examples for independent processes. The general form of an independent process can be written from Eq. (6.5) as,

**Fig. 6.3** Autocorrelation function



$$U_i = \varepsilon_i \quad (6.6)$$

$$y = \text{random ('normal', 5,3,[1 100]);}$$

This is also referred to as the white noise process in the literature. For its simulation, there are a set of PDFs such as Gaussian, Gamma, Weibull, Pearson, Gumbel, log-normal, and others, and their software are available in MATLAB tool box. It is important to mention at this point that more than the PDF, their cumulative distribution functions (CDFs), are used in many application studies including simulation.

Herein, the most common PDF, normal (Gaussian) PDF mathematical form, is given with two parameters, namely the arithmetic average,  $\mu$  and standard deviation,  $\sigma$ , as,

$$f(\varepsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\varepsilon - \mu)^2}{2\sigma^2}\right) \quad (6.7)$$

The shape of this function is similar to the one given in Fig. 6.2 with zero mean,  $\mu = 0$ , and variance  $\sigma$ . The parameter  $\mu$  that is not equal to 1 shifts this figure to the left or right depending on its negative and positive value, respectively, so the arithmetic average may be called as shift parameter. On the other hand,  $\sigma$  is the scale parameter, which is always positive and either narrows or enlarges the figure.

Figure 6.4 represents a sequence of simulation replicate with 100 values out of a Gaussian PDF with mean 5 and standard deviation equal to 3. The stochastic simulation generation model is available in the MATLAB software library, which is given in one line as follows.

The internal (serial) dependence case can be visually examined by plotting any data value against the next, and hence, there are 99 scatter points in Fig. 6.5. It is obvious that the scatter of points does not indicate any mathematically tractable pattern and one can conclude even visually that the data sequence does not have serial correlation, and therefore, the process has completely independent structure.

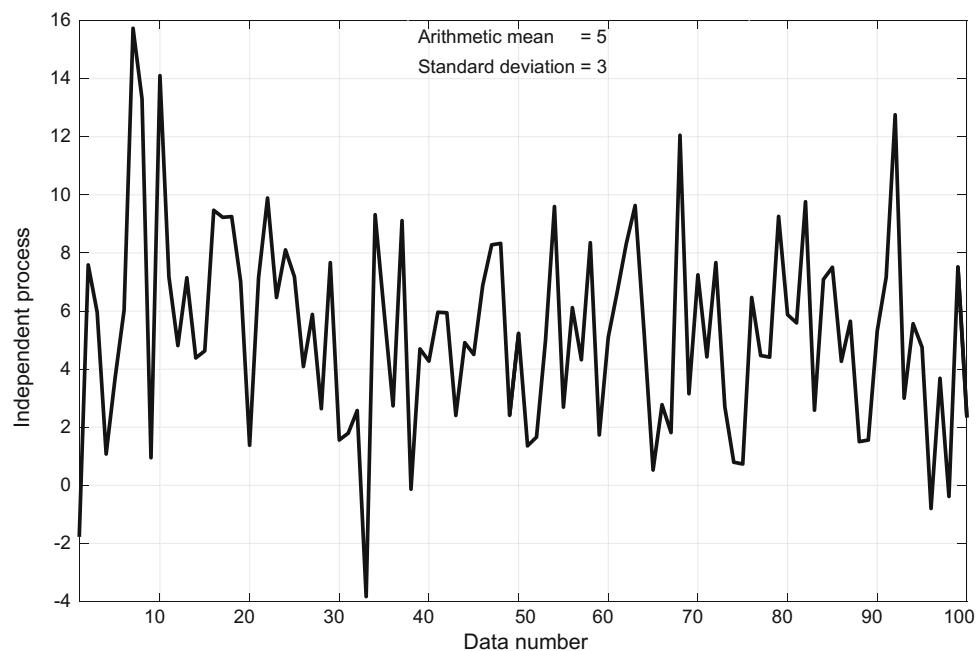
## 6.2.2 Markov Process

The simplest form is the first-order lag process, where only the successive dependences are taken into consideration. This implies addition of a dependence term to Eq. (6.6).

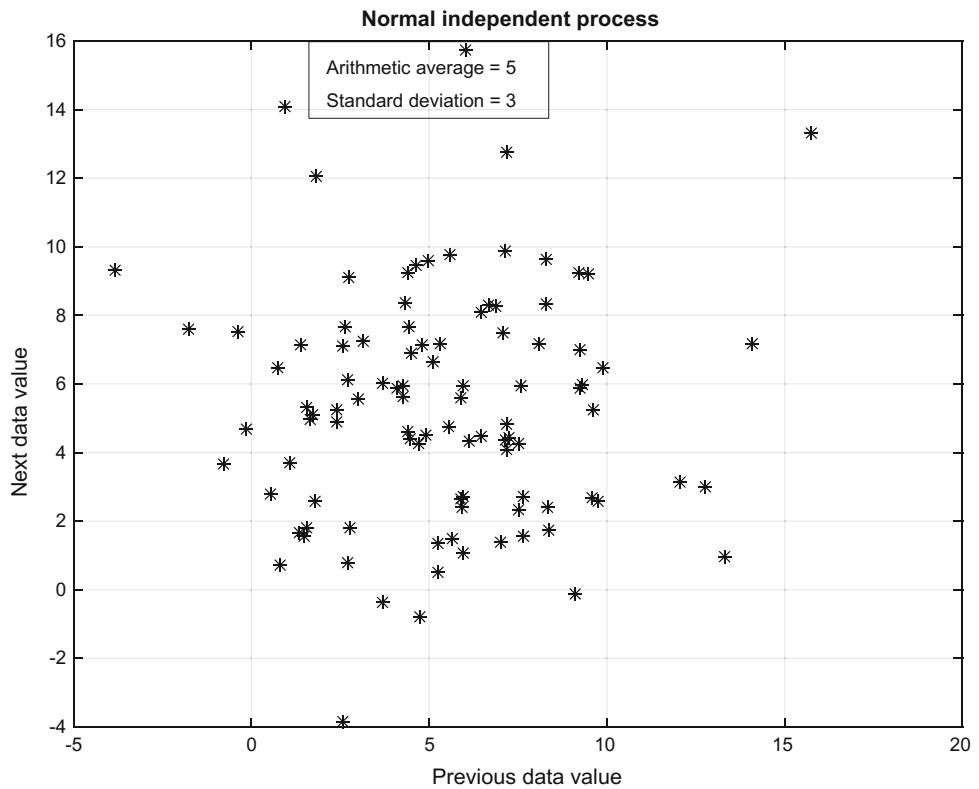
$$X_i = \rho X_{i-1} + \varepsilon_i \quad (6.8)$$

Here,  $\rho$  is the first-order serial correlation coefficient. This is the simplest stochastic process that helps to simulate some short-duration dependence stochastic variables and it is referred to as the first-order Markov or first-order autoregressive process in the literature (Box and Jenkins 1970). The first term on the right-hand side implies that apart from completely random component there is another one, which has some amount of internal structure. In actual applications,  $X_i (i = 1, 2, \dots, n)$  is a sequence of measurements from which random values,  $\varepsilon'_i$ 's must be separated or filtered. This could be

**Fig. 6.4** Normal independent process



**Fig. 6.5** Normal independent processes



achieved if the  $\rho$  value is known before a head. The sequence has also the arithmetic mean,  $\mu$ , and the standard deviation,  $\sigma$ . Inclusion of these parameters in the first-order Markov process Eq. (6.8) takes explicitly the following form.

$$X_i = \mu + \rho(X_{i-1} - \mu) + \sigma\sqrt{1 - \rho^2}\varepsilon_i \quad (6.9)$$

where this time the completely random term,  $\varepsilon$ , has zero mean and unit standard deviation.

The following MATLAB program generates first-order Markov (autoregressive) process for a given set of parameter ( $\mu$ ,  $\sigma$  and  $\rho$ ) values.

```
function [X] = FirstOrderMarkovProcess(Ave, Std, Rho, N)
% This program generates first order Markov process
% Ave = Arithmetic average
% Std = Standard deviation
% Rho = First order correlation coefficient
% N = The number of generation data
% X = The synthetic time series
eps(1,1:N)=randn(1,N);
% Warm up period
X(1,1)=Ave + Rho*(randn - Ave) + Std*sqrt(1 - Rho*Rho)*randn;
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i-1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i-1);
end
% Proper series generation
eps(1,1:N)=randn(1,N);
X(1,1) = X(1,n);
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i-1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i-1);
end
end
```

Figure 6.6 presents some of the replicates that are generated with indicated parameters by employing Eq. (6.9). It is obvious from these figures that as the dependence increases one can visually appreciate more persistence between successive values along the number of data.

The autocorrelation structure of the first-order Markov process has a power function, which can be derived after some stochastic manipulations as follows (Sen 1974).

$$\rho_k = \rho^k \quad (6.10)$$

Similar to the correlation functions given in Fig. 6.2, the use of this last expression with different positive correlation coefficients leads to a set of autocorrelation functions as in Fig. 6.7.

This is a process where the successive values on the scatter diagram as in Fig. 6.8 have some dependence, which is referred to as the serial correlation of different lags.

### 6.2.3 White Markov Process

The purpose of the white Markov process (WMP) is to reduce the original time series serial dependence function

down to almost independent serial structure or up to longer autocorrelation lags than the Markov process (Sen 1974). Let  $Y_t$  represent the first-order Markov process and addition of an independent random (white noise) component,  $\varepsilon_t$ , with zero mean leads to the WMP,  $Z_t$ , as,

$$Z_t = X_t + \gamma \varepsilon_t \quad (6.11)$$

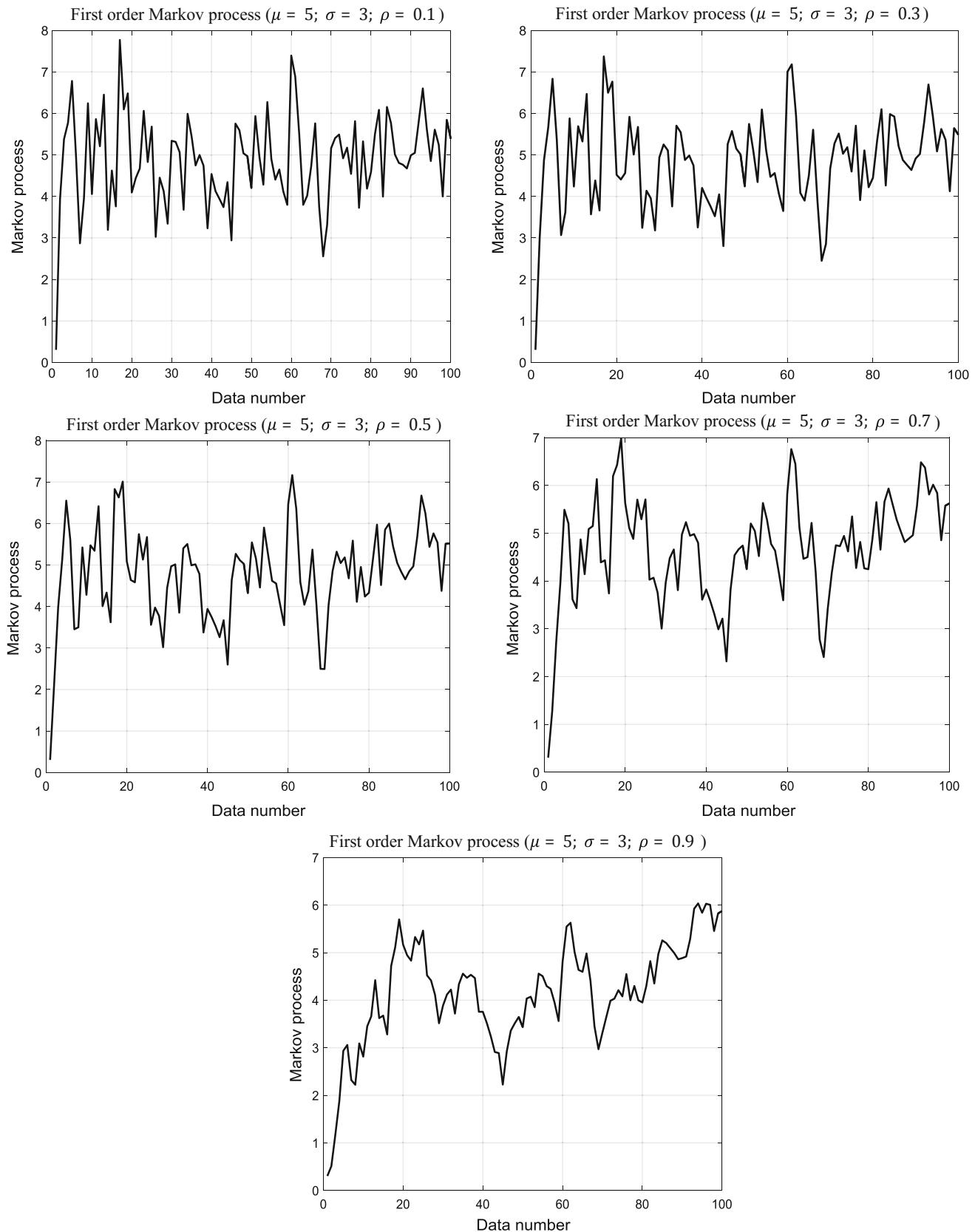
where  $\gamma$  is the standard deviation of the random variable  $\varepsilon_t$  for whitening the Markov process. The lag- $k$  autocorrelation function of the WMP is given by the following expression.

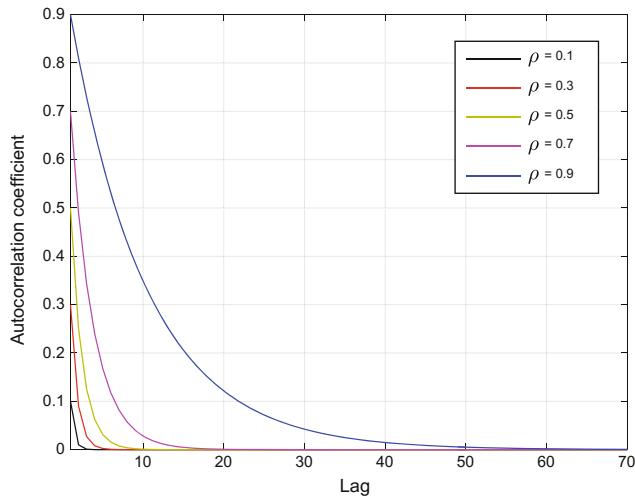
$$\rho_k = \frac{\rho^k}{1 + \gamma^2} = \alpha \rho^k \quad (6.12)$$

Here,  $0 < \alpha < 1$ , and it is referred to as the dependence factor. Hence, the  $k$ th-order autocorrelation function is a function of the lag- $k$  autocorrelation coefficient of the original time series and the independent process standard deviation,  $\gamma$ . Synthetic series generation of WMP is possible by use of the following MATLAB program.

The replicates of WMP by use of Eq. (6.11) with the same set of parameters are presented in Fig. 6.9, which can

```
function [X,Y] = WhiteNoiseProcess(Ave,Std,Rho,gam,N)
% Ave = Arithmetic average
% Std = Standard deviation
% Rho = First order correlation coefficient
% Gam = White noise standard deviation
% N = The number of generation data
% X = The synthetic first order Markov time series
% Y = The synthetic white noise time series
eps(1,1:N)=randn(1,N);
% Warm up period
X(1,1)=Ave + Rho*(randn - Ave) + Std*sqrt(1 - Rho*Rho)*randn;
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i-1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i-1);
end
%Proper series generation
eps(1,1:N)=randn(1,N);
X(1,1) = X(1,N);
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i-1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i-1);
end
% White noise addition
W=random('normal',0,gam,[1,N]);
for i=1:N
    Y(1,i)=X(1,i)+W(1,i);
end
end
```

**Fig. 6.6** Various Markov process scatter diagrams



**Fig. 6.7** Markov process autocorrelation function

be compared with the first-order Markov process correspondences in Fig. 6.6.

Figure 6.10 provides some scatter diagrams between the successive values in the WMP with a set of parameters. The reader may compare these scatter diagrams with the ones in Figs. 6.5 and 6.8 to gain appreciation of the difference between the independent (white), first-order Markov processes and the WMP.

Again the reader may compare these graphs with the ones for the first-order Markov process counterparts in Fig. 6.8.

#### 6.2.4 ARIMA Process

This is also one of the linear autoregressive processes with two components—one is in the form of the first-order Markov process, and the other has the moving average of two successive random components. It has two stochastic model parameters as  $\Phi$  the main indicator of autocorrelation coefficient and  $\Theta$ , which plays the role in the moving average part with some partial effect on the autocorrelation function. The general form of this process is given below.

$$X_i = \Phi X_{i-1} + \Theta \varepsilon_i - \varepsilon_{i-1} \quad (6.13)$$

The autocorrelation function mathematical form of this process can be expressed in terms of its two parameters as (Sen 1974),

$$\rho_1 = \frac{(\Phi - \Theta)(1 - \Phi\Theta)}{1 + \Theta^2 - 2\Phi\Theta} \quad (6.14)$$

and

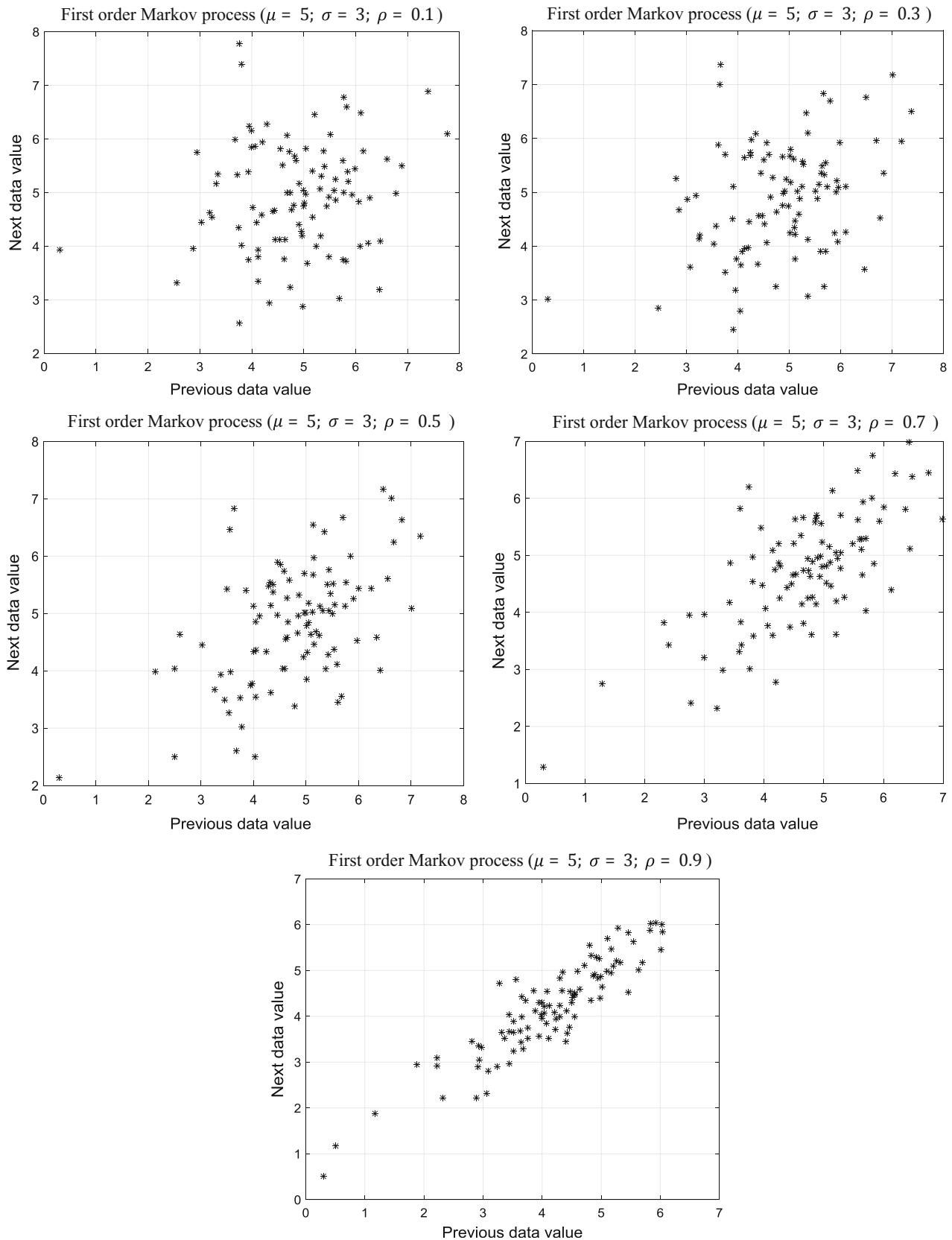
$$\rho_k = \Phi\rho_{k-1} \quad \text{for } k \geq 2 \quad (6.15)$$

On the other hand by taking into consideration the arithmetic mean,  $\mu$ , and the standard deviation,  $\sigma$ , of the measurement time series, Eq. (6.13) can be written as follows.

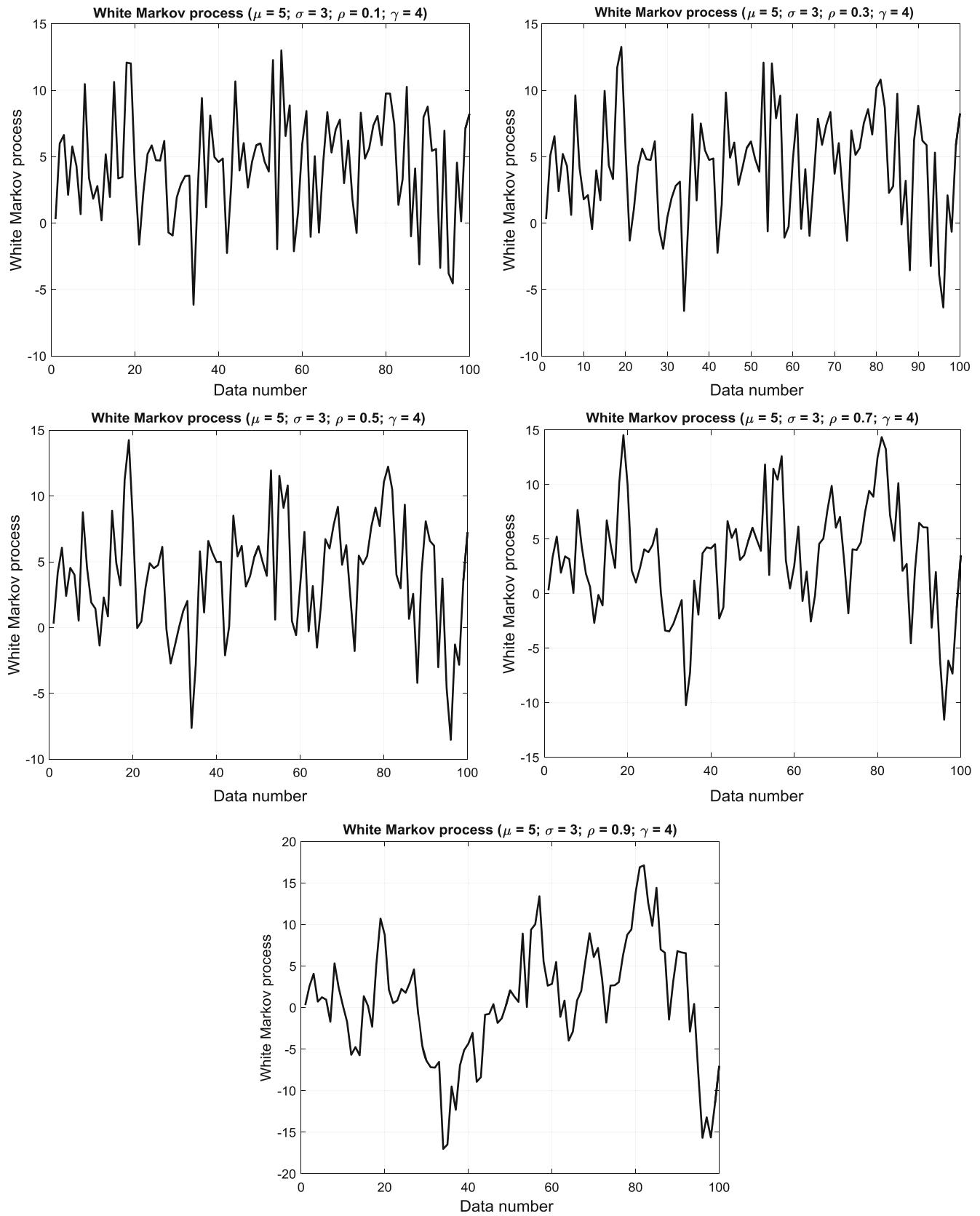
$$(X_i - \mu) = \Phi(X_{i-1} - \mu) + \sigma_s \sigma_\varepsilon (\varepsilon_i - \Theta \varepsilon_{i-1}) \quad (6.16)$$

where

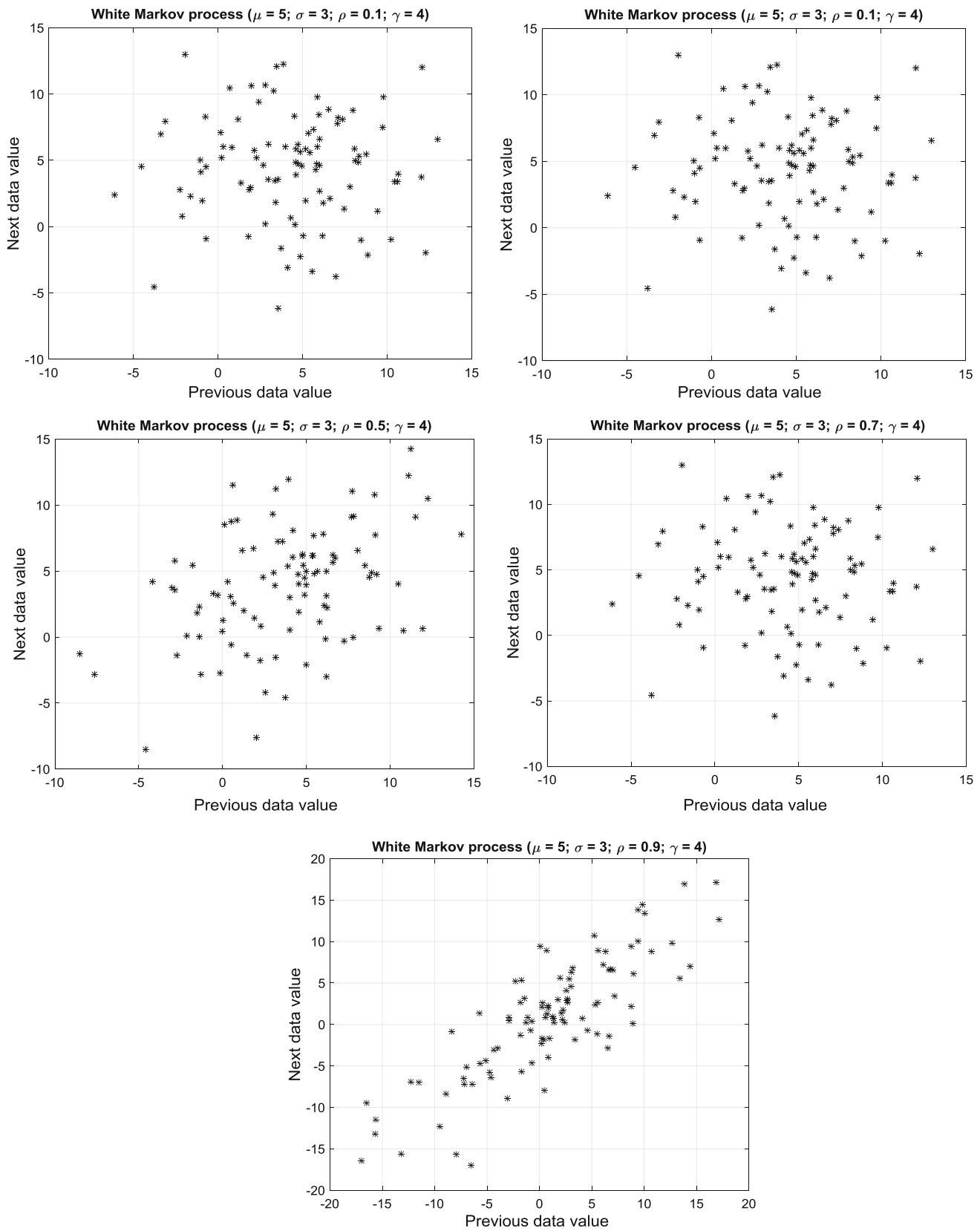
$$\sigma_\varepsilon^2 = \frac{(1 - \Phi^2)}{1 + \Theta^2 - 2\Phi\Theta} \quad (6.17)$$



**Fig. 6.8** Various Markov process scatter diagrams



**Fig. 6.9** White Markov process time series

**Fig. 6.10** White Markov process scatter diagrams

The following MATLAB language written program provides opportunity to generate ARIMA(1,0,1) process with a given set of parameter ( $\Phi$  and  $\Theta$ ) values. A sample

of replicate for the ARIMA(1,0,1) process is presented in Fig. 6.11 with the same model parameters.

```

function [Rho] = ARIMA101Process(X,Phi,Theta)
% This program calculates the autocorrelation function of ARIMA(1,0,1)process
% X is two dimensional array measurement time series (TimeIntervalxData)
% Phi is one of the ARIMA(1,0,1) process parameters
% Theta is the next parameter of the ARIMA(1,0,1) process
%
n=length(X(:,1));
n1=n-1;
m2=mean(X(:,2));
s2=std(X(:,2));
Rho(1)=(Phi-Theta)/(1+Theta^2-Phi*Theta);
for i=2:50
    Rho(i)=Phi*Rho(i-1);
end
MH=max(Rho);
figure
plot(Rho,'LineWidth',2,'Color','k')
grid on
box on
xlabel('Lag')
ylabel('Autocorrelation coefficient')
title('ARIMA(1,0,1) process autocorrelation function')
text(5,0.9*MH,['Phi = ',num2str(Phi),' Theta = ',num2str(Theta)])
epic=random('normal',0,1,[1 n]);
sepia=std(epic);
S(1)=X(1);
for i=2:n
    S(i)=m2+Phi*(S(i-1)-m2)+s2*sepia*(Theta*(epic(i)-epic(i-1)));
end
figure
MS=min(S);
M1=max(S);
m1=min(X(:,1));
M1=max(X(:,1));
plot(X(:,1),S,'LineWidth',2,'Color','k');
grid on
box on
xlabel('Year')
ylabel('Values')
title('ARIMA(1,0,1) process simulation sequence')
text(1.01*m1,0.9*M1,['Phi = ',num2str(Phi),' Theta = ',num2str(Theta)])
figure
scatter(S(1:n1),S(2:n),'*')
grid on
box on
xlabel('Previous data value')
ylabel('Next data value')
title('ARIMA(1,0,1) process')
text(1.01*mS,0.98*MS,['Phi = ',num2str(Phi),' Theta = ',num2str(Theta)])
end

```

One can compare this graph with the ones in Figs. 6.6 and 6.9 so as to appreciate the distinction among this, first-order Markov and white Markov processes. Finally, a sample set of autocorrelation functions is given in Fig. 6.12 for the ARIMA process.

The successive value scatter diagram for the ARIMA(1,0,1) process is given in Fig. 6.13 with  $\Phi = 0.8$  and  $\Theta = 0.5$ . Its comparison with Figs. 6.8 and 6.10 exposes distinctive character of the ARIMA (1,0,1) process.

In Fig. 6.14 is the ARIMA process simulation replicate with indicated model parameters.

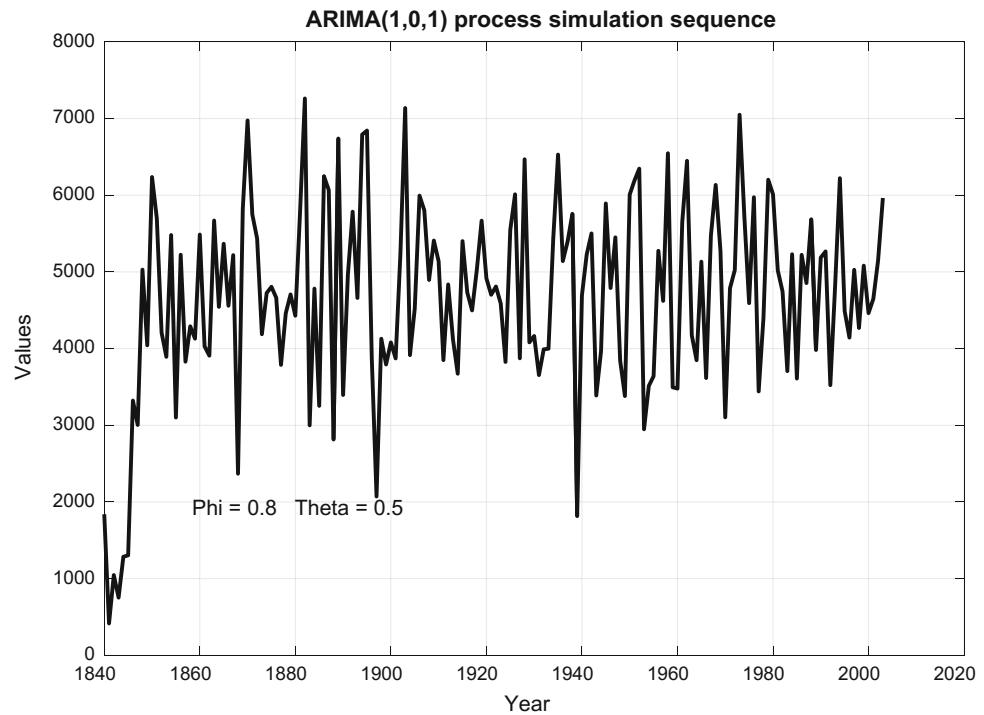
### 6.3 Maximum Rainfall Simulation

Recently, there are many incidences all over the world that extreme rainfall events take place rather suddenly leading to floods, flash floods, and consequent inundation areas, where urban centers, settlement areas, militarily, agriculturally and touristically important areas are hit. Most often these incidences are interpreted as the consequence of present global warming, climate change as a result of greenhouse emissions

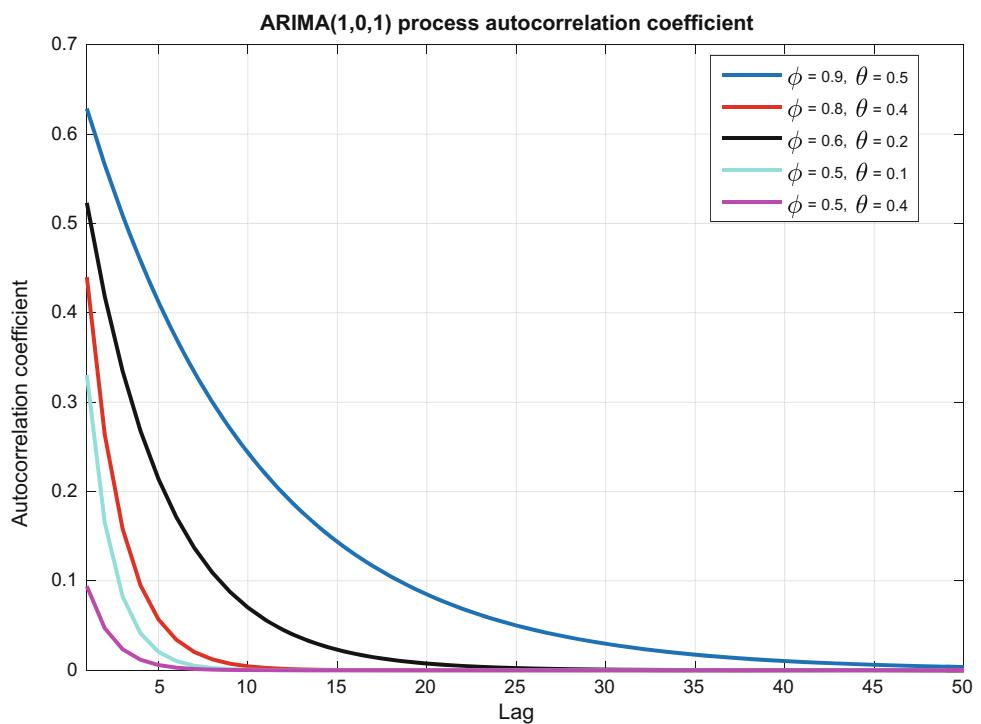
into the atmosphere. This section provides quantitative and qualitative information along two directions concerning the maximum rainfall simulation.

- (1) The climate change impact in the annual daily maximum rainfall amounts is searched in terms of increasing, neutral, and decreasing trends (Chap. 10). In this direction, general circulation model (GCM) scenario results (Chap. 7) are not taken into consideration for two reasons,
  - (a) The GCM model scenario results need to be downscaled. Although there are regional climate change software of different types, their internal structure is rather complicated and they are more suitable for research purposes, whereas in practical project applications rather simpler, reliable, and easily applicable friendly based software must be developed by consideration of dynamic or statistical downscaling procedure,
  - (b) Prior to a general downscaling study, it is necessary to identify possible past climate change impacts on the available records. Here the main

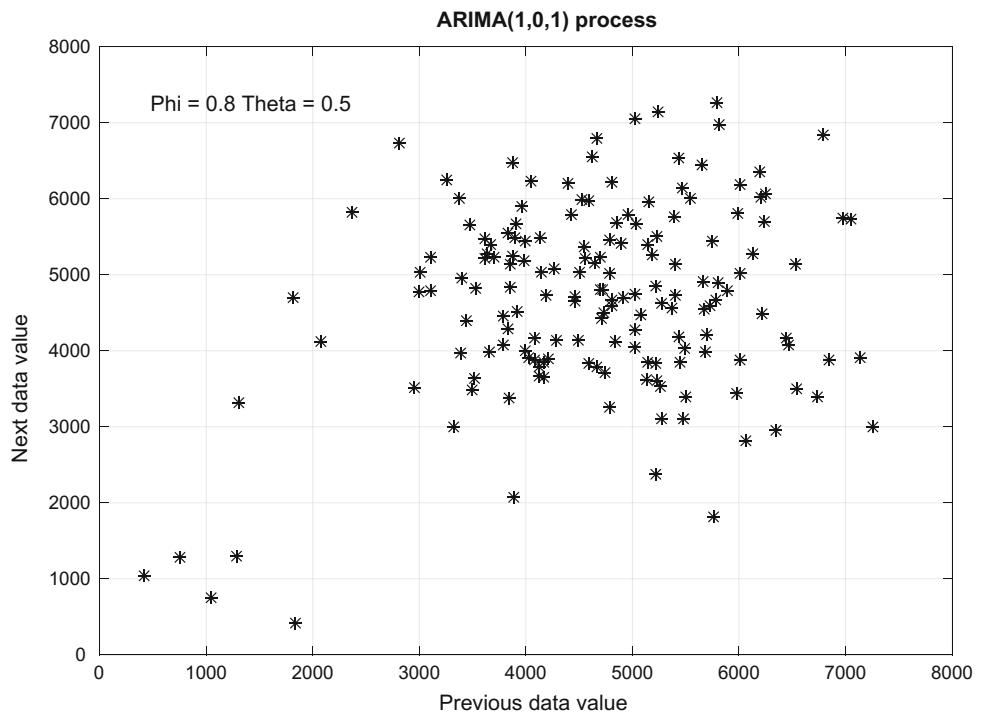
**Fig. 6.11** ARIMA(1,0,1) process time series



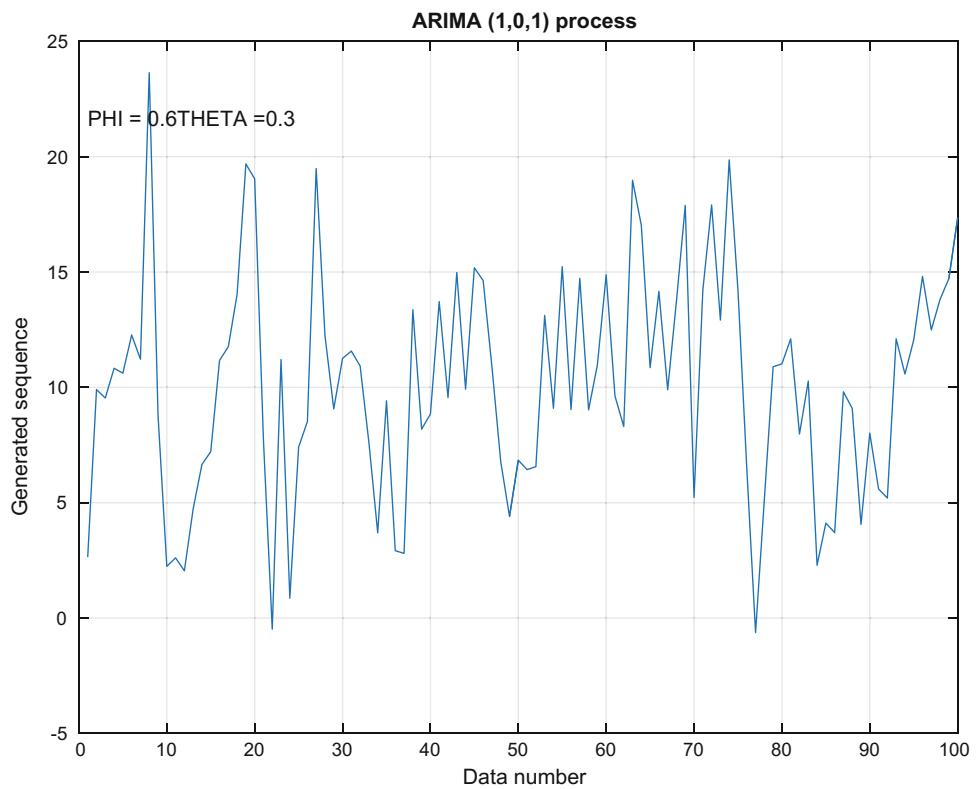
**Fig. 6.12** ARIMA process autocorrelation function



**Fig. 6.13** ARIMA(1,0,1) process scatter diagram



**Fig. 6.14** ARIMA (1,0,1) process simulation replica



question is whether there are significant differences in each record period when the last half is compared with the first half. Such an innovative study has been presented into the literature by Sen (2012, 2014, 2017), and it is being widely applicable all over the world for the preliminary investigation of climate change effects (Chap. 10).

- (2) The second most important direction is to extend the available historical records to future periods, this has been achieved by software written for Saudi Geological Survey by the author in MATLAB language, and the records are extended up to 2030 on the basis of the most convenient cumulative distribution functions (CDFs) as already mentioned in Chap. 2.

```

function [Y, MX, SX, CX, MY, SY, CY] = AnnualMaxRainSimulation(X, IY, FY, n, Title)
% This program is written by Zekai Sen on 21 September 2015 for extension
% of annual maximum rainfall records through simulation
% X      : The annual maximum rainfall time series
% Y      : Simulation time series
% m      : Length of the simulation time series
% IY     : Initial year
% FY     : Final year
% n = 1 Gamma probability distribution fit
% n = 2 Log-normal probability distribution fit
% n = 3 Extreme value (Gumbel) probability distribution fit
% n = 4 Generalized extreme value (Pearson) probability distribution fit
N=length(X);
N2=floor(N/2);
Year(1)=IY;
m=FY-IY; % Simulation duration
X1=sort(X(1:N2));
X2=sort(X(N2+1:N));
SX=2*(mean(X2)-mean(X1))/N; % Trend slope in the given time series
Xmin=min(X);
XMax=max(X);
XMean=mean(X);
TMean=N/2;
aX=XMean-SX*TMean;
xi=IY;
xf=IY+N-1;
yi=aX;
yf=aX+SX*(xf-xi);
figure
plot(IY:1:IY+N-1,X,'LineWidth',2,'Color','k')
title>Title
xlabel'Years'
ylabel'Annual maximum rainfall (mm)'
grid on
hold on
line([xi xf],[yi yf],'LineWidth',2,'Color','k');
text(15,XMax-2,['Intercept = ' num2str(aX), ' Slope = ' num2str(SX)])
for i=1:N
    T(i)=aX+SX*i; % Trend component time series
end
XT=X-T; % Trend free time series
XTMean=mean(XT);
XTStDev=std(XT);
XS=(XT-XTMean)/XTStDev; % Standard time series
c=correlation(XS, 6);
XTCor=c(1);
if n==1
    XX=X(X>0); % Take only positive numbers
    P=gamfit(XX);
    r=random('gamma',P(1),P(2),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Gamma PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
elseif n==2
    XX=X(X>0);
    P=lognfit(XX);
    r=random('logn',P(1),P(2),[100 1]);
    MG=mean(r);

```

.....Continued.....

```

SG=std(r);
r=(r-MG)/SG;
text(15,XMax-(XMax-Xmin)*0.1,'Log-normal PDF ')
text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
elseif n==3
    XX=X(X>0);
    P=evfit(XX);
    r=random('ev',P(1),P(2),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Extreme value (Gumbel) PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
else
    XX=X(X>0);
    P=gevfit(XX);
    r=random('gev',P(1),P(2),P(3),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Generalized extreme value (Pearson) PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2)), 'Gamma = ' num2str(P(3))])
end

% Dependent time series generation
YG(1)=r(1);
for i=2:m
    YG(i)=XTCor*YG(i-1)+r(i);
end
YS=YG*XTStDev+XTMean;
xf=IY+m-1;
yf=aX+SX*(xf-xi);
for i=1:m
    TS(i)=aX+SX*i; % Trend component time series
end
Y=TS+YS;
plot(IY:1:IY+m-1,Y,'Color','r')
%plot(TS,'LineWidth',2,'Color','r')
line([xi xf],[yi yf],'LineWidth',2,'Color','r');
MX=mean(X);
SX=std(X);
CX=c(1);
MY=mean(Y(1:m));
SY=std(Y(1:m));
CY=correlation(Y(1:m),1);
legend('Historic daily maximum rainfall (mm)', 'Historic trend', 'Generated daily maximum
rainfall (mm)', 'Future trend')
end

XT=X-T'; % Trend free time series
XTMean=mean(XT);
XTStDev=std(XT);
XS=(XT-XTMean)/XTStDev; % Standard time series
c=correlation(XS,6);
XTCor=c(1);
if n==1
    XX=X(X>0); % Take only positive numbers
    P=gamfit(XX);
    r=random('gamma',P(1),P(2),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;

```

.....Continued.....

```

text(15,XMax-(XMax-Xmin)*0.1,'Gamma PDF ')
text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
elseif n==2
    XX=X(X>0);
    P=lognfit(XX);
    r=random('logn',P(1),P(2),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Log-normal PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
elseif n==3
    XX=X(X>0);
    P=evfit(XX);
    r=random('ev',P(1),P(2),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Extreme value (Gumbel) PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2))])
else
    XX=X(X>0);
    P=gevfit(XX);
    r=random('gev',P(1),P(2),P(3),[100 1]);
    MG=mean(r);
    SG=std(r);
    r=(r-MG)/SG;
    text(15,XMax-(XMax-Xmin)*0.1,'Generalized extreme value (Pearson) PDF ')
    text(15,XMax-(XMax-Xmin)*0.2,['Alpha = ' num2str(P(1)), ' Beta = ' num2str(P(2)), 'Gamma = ' num2str(P(3))])
end
% Dependent time series generation
YG(1)=r(1);
for i=2:m
    YG(i)=XTCor*YG(i-1)+r(i);
end
YS=YG*XTStDev+XTMean;
xf=IY+m-1;
yf=aX+SX*(xf-xi);
for i=1:m
    TS(i)=aX+SX*i; % Trend component time series
end
Y=TS+YS;
plot(IY:1:IY+m-1,Y,'Color','r')
%plot(TS,'LineWidth',2,'Color','r')
line([xi xf],[yi yf],'LineWidth',2,'Color','r');
MX=mean(X);
SX=std(X);
CX=c(1);
MY=mean(Y(1:m));
SY=std(Y(1:m));
CY=correlation(Y(1:m),1);
legend('Historic daily maximum rainfall (mm)', 'Historic trend', 'Generated daily maximum rainfall (mm)', 'Future trend')
end

```

The following points are important for those who would like to implement the information given in this section.

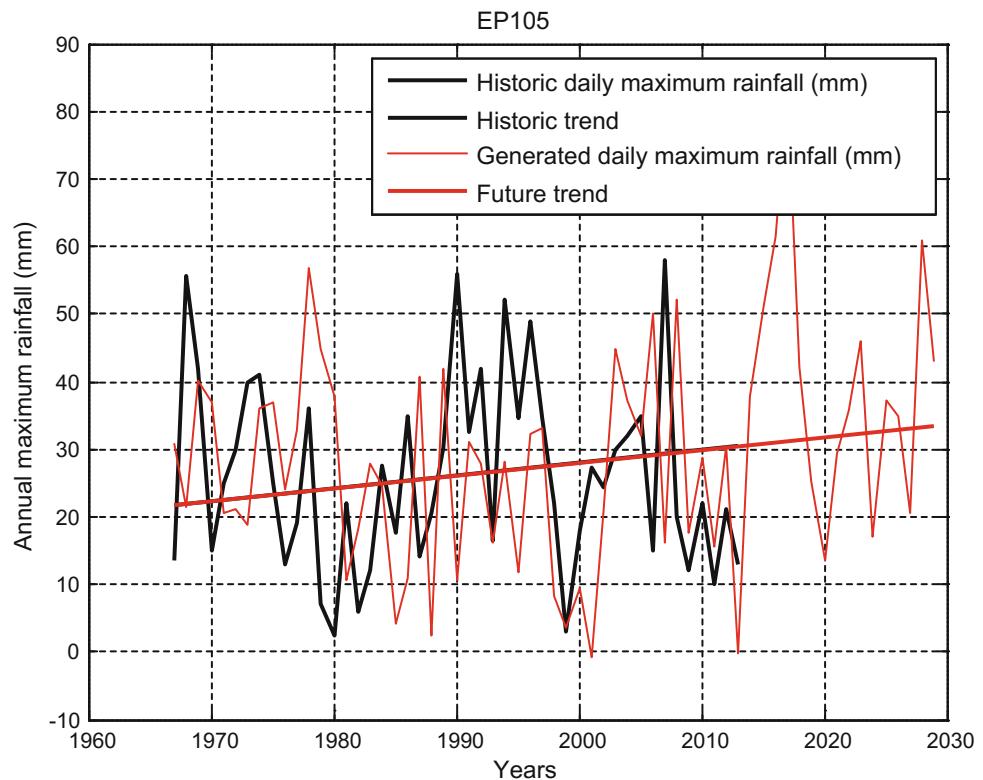
- (1) In cases of neutral and decreasing trend situations at any meteorology station, the future planning of any engineering structure design in a drainage basin can continue according to classical methodologies,
- (2) In cases of increasing trend, the amount of design variable can be augmented depending on the trend slope according to the following risk,  $R$ , formulation given by Sen et al. (2017).

$$R = \frac{1 + \alpha}{T} \quad (6.18)$$

Here,  $T$  is the return period and  $\alpha$  is the slope of increasing trend. For  $\alpha = 0$  the classical risk formulation is valid as explained in Chap. 2. The application of the above MATLAB simulation software for data at the Kingdom of Saudi Arabia yields Fig. 6.15.

In the following another simulation program in MATLAB is given for monthly time series simulation generations. Also subsequent MATLAB program, [cor] = correlation(X, ml), is for autocorrelation coefficient calculations called by the main program.

**Fig. 6.15** EP105 station simulation (extension) time series



```

function [G,mu,su,cu] = MonthlySeriesGeneration(X)
% This program is written on 23 July 2014 by Zekâi Sen.
% This program generates synthetic sequences according to the correlations between two months
% X = Monthly data matrix(year i.e. nx12)
% ny = Number of years
% N = YearxMonth number
% G = Generated series
% m, s, (mu, su) - Observation (Generation) arithmetic average and standard deviations
% c (cu) = Observation (Generation) monthly correlation coefficients
ny=length(X(:,1));
N=ny*12;
% Calculate monthly arithmetic averages and standard deviations
m=mean(X);
s=std(X);
% Observation series successive monthly correlation coefficient calculations
for i=1:11
    cc=corrcoef(X(:,i),X(:,i+1));
    c(i)=cc(1,2);
end
cc=corrcoef(X(:,12),X(:,1));
c(12)=cc(1,2);
% Calculation of standardized monthly series
for i=1:12
    for j=1:ny
        x(j,i)=(X(j,i)-m(i))/s(i);
    end
end
% Find generation error sequence
hata=random('normal',0,1,[1,N]);
% Monthly standard error matrix
k=0;
for i=1:ny
    for j=1:12
        k=k+1;
        h(i,j)=hata(k);
    end
end
% Convert independent monthly values to dependent values
% As the first monthly sequence adapt the first month's values
z(1:ny,1)=x(1:ny,1);
% Generate other monthly sequences
for j=2:12
    for i=1:ny
        z(i,j)=c(j)*z(i,j-1)+h(i,j);
    end
end
% Find monthly generated sequence matrix
for j=1:12
    for i=1:ny
        G(i,j)=z(i,j)*s(j)+m(j);
    end
end
% Calculate the statistical parameters of generation monthly sequence
mu=mean(G);
su=std(G);
% Calculate monthly sequential series correlation coefficients
for i=1:11
    ccc=corrcoef(G(:,i),G(:,i+1));
    cu(i)=ccc(1,2);
end
ccc=corrcoef(G(:,11),G(:,12));
cu(12)=ccc(1,2);
end
.....Continued.....

```

```
% Calculate successive monthly correlation coefficients
for i=1:11
    ccc=corrcoef(G(:,i),G(:,i+1));
    cu(i)=ccc(1,2);
end
ccc=corrcoef(G(:,11),G(:,12));
cu(12)=ccc(1,2);
figure
plot(X)
title('Title')
xlabel('Years')
ylabel('Natural monthly total rainfall (mm)')
figure
plot(abs(G))
title('Title')
xlabel('Years')
ylabel('Synthetic monthly total rainfall (mm)')
end
```

## 6.4 Computer Programs

The following short and simple “correlation” sub-routine function program is called by the two above MATLAB programs.

Simulation is a process that can generate a set of replicates as much as desirable number. The question is how to benefit from this bunch of replicates? The following MATLAB program takes the arithmetic averages of all generated sequence and hence results in a single representative replicate.

```
function [cor]=correlation(X,m1)
% This program is written by Zekai Sen in 1975 in Fortran language
% x is the time series vector
% m1 is the maximum lag
n=max(size(X));
m=mean(X);
s=std(X);
for i=1:n
    y(i)=(X(i)-m)/s;
end
for i=1:m1
    sum=0;
    ne=n-i;
    for j=1:ne
        sum=sum+y(j)*y(i+j);
    end
    cor(i)=sum/ne;
end
plot(cor)
end
```

```

function [GX] = EnsembleGeneration(X,Title,NGY)
% This program is written by Zekai Sen on 23 March 2015.
% X = Monthly data matrix (year i.e., nx12)
% NGY = Number of generation year: NOTICE ??? NGY must not be taken as 1
% GX = Generation time series
% gm, gs, gc are the arithmetic average, standard deviation and correlation coefficient of
% generation time series.
ny=length(X(:,1));
for l=1:NGY
    [U,mu,su,cu] = MonthlyFirstOrderGeneration(X,Title);
    UU(1,1:ny,1:12)=U(1:ny,1:12);
    muu(1,1:12)=mu(1,1:12);
    suu(1,1:12)=su(1,1:12);
    cuu(1,1:12)=cu(1,1:12);
end
% Calculates average generated monthly values
for i=1:12
    summ=0;
    sums=0;
    sumc=0;
    for j=1:ny
        k=0;
        sumx=0;
        for l=1:NGY
            sumx=sumx+UU(l,j,i);
            summ=summ+muu(l,i);
            sums=sums+suu(l,i);
            sumc=sumc+cuu(l,i);
        end
        GX(j,i)=sumx/NGY;
        gm(i)=summ/NGY;
        gs(i)=sums/NGY;
        gc(i)=sumc/NGY;
    end
end
end

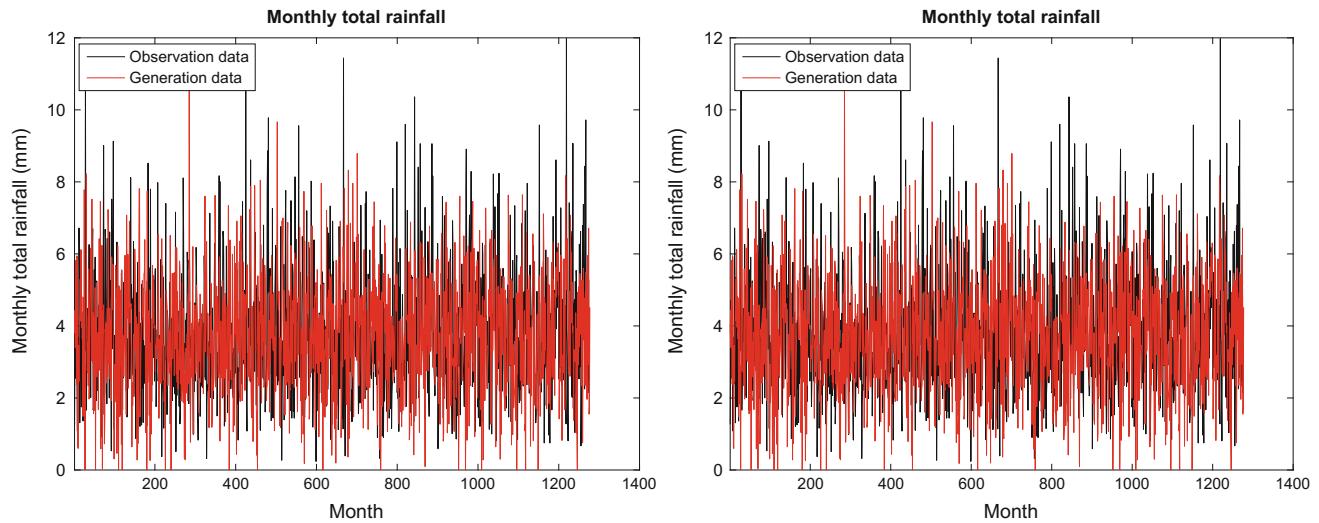
```

## Appendix

Figure 6.16 presents the output from the above MATLAB programs the observation sequence with the ensemble average generation data. It is visually obvious that the observation and generation time series have more or less similar statistical parameters; i.e., they are statistically indistinguishable.

## Seasonal Generation Process

The first-order Markov synthetic time series simulation methodology assumes that the given time series has a stationarity structure, which means that the statistical parameters are constant over the whole time series duration. Examples for such time series are annual records of natural events that are not affected by the astronomical and seasonal



**Fig. 6.16** Observation and generation time series

effects. However, daily, weekly and monthly records shorter than annual durations each time series has inborn seasonal effects. The best example for such series is monthly records of natural phenomena such as hydro-meteorological events.

In the case of seasonality, the statistical parameters vary according to months as explained in Chap. 2, Sect. 2.8. Their simulations need a stochastic process with monthly arithmetic average and standard deviations as in Table 2.7 (Chap. 2). In the same section, the elimination of arithmetic average and the standard deviation procedure has been explained. In this manner, the monthly time series is converted to weak stationarity, whereby monthly arithmetic averages are equal to zero and the standard deviation is equal to one. However, this time series still have non-seasonality in correlation coefficient between two successive months, which is difficult to eliminate by simple statistical methodologies.

It is necessary to develop a simulation process, which is similar to first-order Markov process, but with variable

statistical parameters and successive monthly correlation coefficients. In the following is the monthly counterpart of the stochastic model in Eq. (6.9) as follows.

$$X_i = \mu_i + \rho_{i,i-1}(X_{i-1} - \mu_i) + \sigma_i \sqrt{1 - \rho_{i,i-1}^2} \epsilon_i \quad (6.19)$$

$$(i = 1, 2, \dots, 12)$$

where  $X_i$  is the monthly value,  $\mu_i$  and  $\sigma_i$  are the monthly arithmetic average and the standard deviation, respectively, and  $\rho_{i,i-1}$  is the correlation coefficient between successive  $i$  and  $i - 1$  months ( $i = 1, 2, \dots, 12$ ), and finally,  $\epsilon_i$  is an independent normal random variable that accords with the normal (Gaussian) PDF with zero mean and unit variance.

The MATLAB program of this monthly simulation stochastic process is given in the following box. In the hydro-meteorology literature, this model is referred to as the Thomas–Fiering stochastic process for monthly variable generations (Thomas and Fiering 1962).

```

function [U,m,s,c,mu,su,cu] = ThomasFiering(X,ny)
% This program generates synthetic sequences according to Thomas-Fiering method
% X = Monthly observation data (year i.e. nx12)
% n = Observation year number
% ny = Generation data year number
% nu = Generation year number
% U = Generation series
% c = Monthly correlation between successive observation series
% cu = Monthly correlation between successive generation series
% Calculation of monthly arithmetic average and standard deviation
n=length(X(:,1));
N=ny*12;
m=mean(X);
s=std(X);
for i=1:11
    cc=corrcoef(X(:,i),X(:,i+1));
    c(i)=cc(1,2);
end
cc=corrcoef(X(:,12),X(:,1));
c(12)=cc(1,2);
% Standardized monthly data
for i=1:12
    for j=1:n
        x(j,i)=(X(j,i)-m(i))/s(i);
    end
end
% Measurement error terms in a series
k=0;
for i=1:n
    for j=1:12
        k=k+1;
        H(k)=x(i,j); % Measurement error series
    end
end
% Find generation error terms
hata=random('normal',0,1,[1,N]);
k=0;
nu=N/12;
for i=1:nu
    for j=1:12
        k=k+1;
        h(i,j)=hata(k);
    end
end
% Convert independent monthly series to dependent series
for j=1:11
    for i=1:nu
        h(i,j+1)=c(j)*h(i,j);
    end
end
h(1:end,1)=c(12)*h(1:end,12);
% Calculation year number for generation
nu=N/12;
% Find monthly generation series
k=0;
for i=1:nu
    for j=1:12
        k=k+1;
        U(i,j)=hata(k)*s(j)+m(j);
    end
end
mu=mean(U);
su=std(U);
for i=1:11
    ccc=corrcoef(U(:,i),U(:,i+1));
    cu(i)=ccc(1,2);
end
ccc=corrcoef(U(:,11),U(:,12));
cu(12)=ccc(1,2);
end

```

In the following is another MATLAB language written program for generation of seasonal time series for simulation purpose.

```

function [U,mu,su,cu] = MonthlyFirstOrderGeneration(X,Title)
% This program is written by Zekai Sen on 23 July 2014
% It generates synthetic sequences between the successive months
% X = Monthly time series(Year i.e. nx12)
% ny = Number of observation years
% N = Number of YearXMonth
% U = Generation sequence
% m, s, (mu, su) - Observation (Generation) sequence arithmetic average,
% and standard deviation values
% c (cu) = Observation (Genaration) monthly sequence correlations
ny=length(X(:,1));
N=ny*12;
% Calculate monthly mean and standard deviation
m=mean(X);
s=std(X);
%Calculate successive monthly correlations
for i=1:11
    cc=corrcoef(X(:,i),X(:,i+1));
    c(i)=cc(1,2);
end
cc=corrcoef(X(:,12),X(:,1));
c(12)=cc(1,2);
% Calculate monthly standard data
for i=1:12
    for j=1:ny
        x(j,i)=(X(j,i)-m(i))/s(i);
    end
end
% Find generation error terms
hata=random('normal',0,1,[1,N]);
% Find the standard sequence
k=0;
for i=1:ny
    for j=1:12
        k=k+1;
        h(i,j)=hata(k);
    end
end
% Convert independent monthly series to dependent forms
% As for the first monthly series take the 1. month observation series
z(1:ny,1)=x(1:ny,1);
% Generate other monthly series
k=0;
for i=1:ny
    for j=2:12
        k=k+1;
        xx(k)=X(i,j);
        z(i,j)=c(j)*z(i,j-1)+h(i,j);
    end
    z(i,1)=c(1)*z(i,12)+h(i,1);
end
% Find generated series
figure
plot(xx,'k')
xlabel('Month')
ylabel('Monthly total rainfall (mm)')
title('Monthly total rainfall')
hold on
k=1;
for i=1:ny
    for j=2:12
        k=k+1;
        .....Continued.....

```

```

U(i,j)=z(i,j)*s(j)+m(j);
uu(k)=U(i,j);
if uu(k) < 0
    uu(k)=0;
else
end
end
U(i,1)=z(i,1)*s(1)+m(1);
uu(1)=U(i,1);
if uu(1) < 0
    uu(1)=0;
else
end
end
plot(uu,'r')
legend('Observation data','Generation data','Location','Northeast')
% Calculate the statistical parameters of generation time series
mu=mean(U);
su=std(U);
% Calculate successive monthly correlation coefficients
for i=1:11
    ccc=corrcoef(U(:,i),U(:,i+1));
    cu(i)=ccc(1,2);
end
ccc=corrcoef(U(:,12),U(:,1));
cu(12)=ccc(1,2);
figure
plot(X)
grid on
box on
title>Title)
xlabel'Years')
ylabel'Natural monthly total rainfall (mm)')
legend'January','February','March','April','May','June','July','August',...
    'September','October','November','December')
figure
plot(abs(U))
grid on
box on
title>Title)
xlabel'Years')
ylabel'Synthetic monthly total rainfall (mm)')
legend'January','February','March','April','May','June','July','August',...
    'September','October','November','December')
figure
plot(m,'k','LineWidth',2)
hold on
plot(mu,'r','LineWidth',2)
title'Monthly arithmetic averages')
xlabel'Month')
ylabel'Arithmetic average')
grid on
box on
legend'Observation values','Genaration values','NorthEast')
figure
plot(s,'k','LineWidth',2)
hold on
plot(su,'r','LineWidth',2)
title'Monthly standard deviations')
xlabel'Month')
ylabel'Standard deviations')

```

.....Continued.....

```

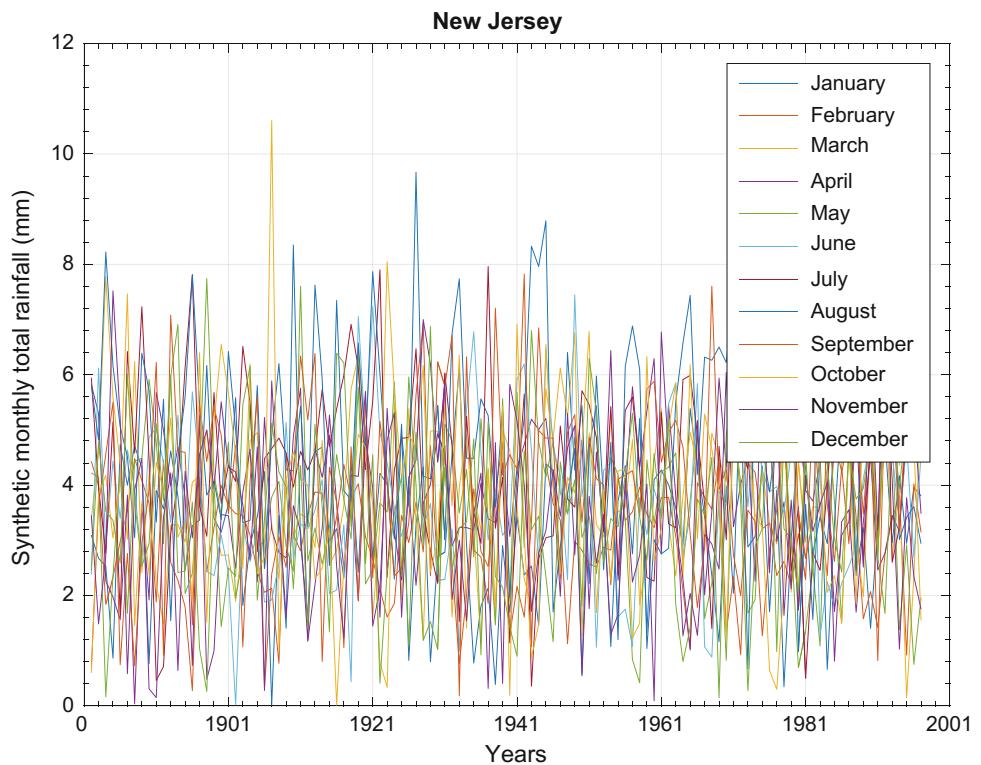
grid on
box on
legend('Observation values','Genaration values','NorthEast')
figure
plot(c,'k','LineWidth',2)
hold on
plot(cu,'r','LineWidth',2)
title('Inter-monthly correlation coefficients')
xlabel('Month')
ylabel('Inter-monthly correlation coefficients')
grid on
box on
legend('Observation values','Genaration values','NorthEast')
end

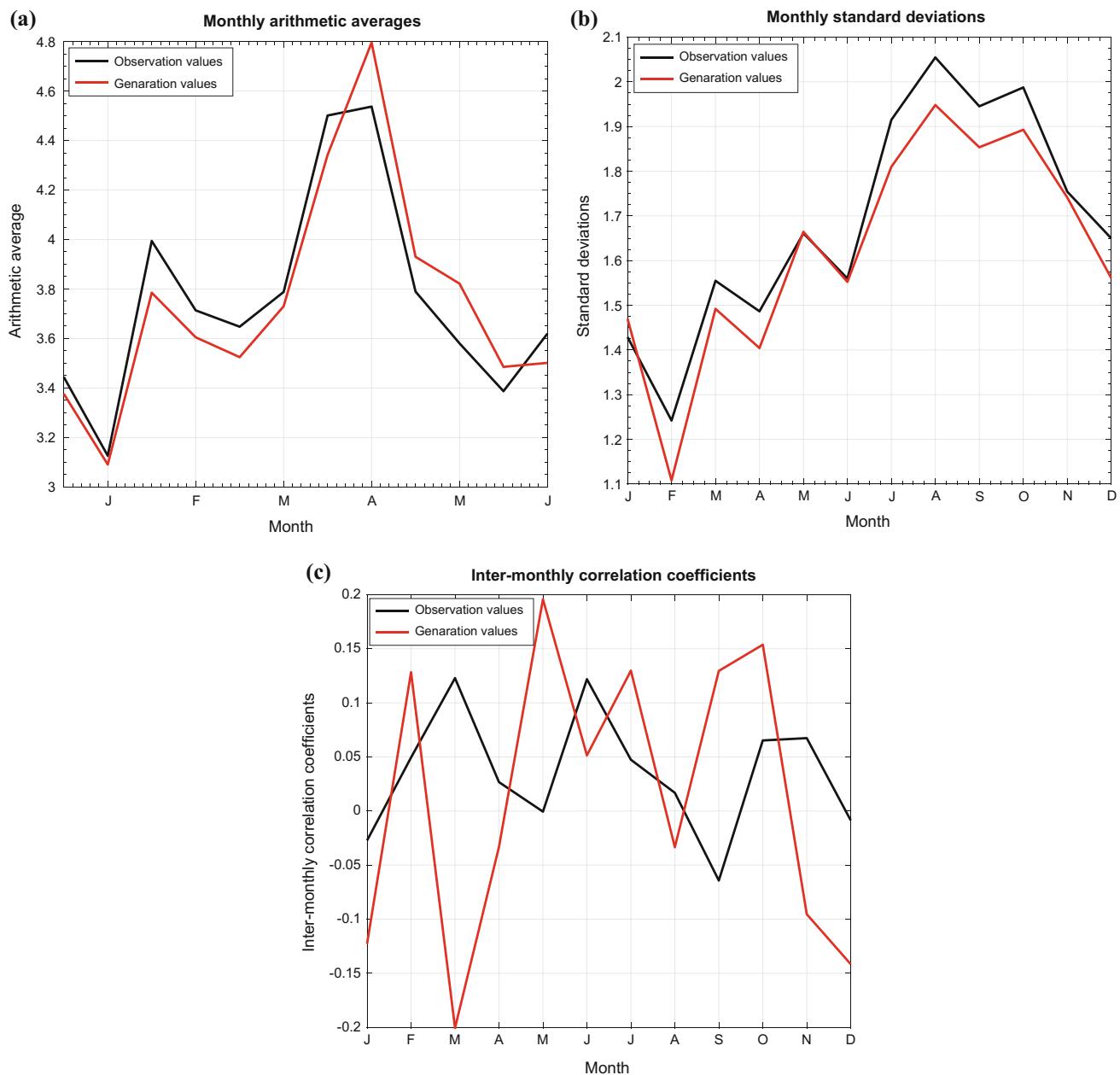
```

On the other hand, Fig. 6.17 shows synthetic sequence generation of monthly rainfall amounts for 12 months, which are also generated by the aforementioned MATLAB programs.

In order to indicate that the monthly statistics (arithmetic averages, standard deviations, and successive monthly correlation coefficients), similarity between observed and generation values Fig. 6.18 is presented.

**Fig. 6.17** Monthly total rainfall simulation traces





**Fig. 6.18** Observation and generation sequences: **a** arithmetic averages, **b** standard deviations, **c** correlation coefficients

## References

- Box, G.E.P., and Jenkins, G.M., (1970). Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco.  
 Cox, D.R., and Miller, H.D., (1965). The Theory of Stochastic Processes Methuen, London.  
 Sen, Z., (1974). Small sample properties of the stochastic processes and the Hurst phenomenon in hydrology. Unpublished Ph. D. Thesis, University of London, Imperial College of Science and Technology, 296 pp.  
 Sen, Z., (2012). Innovative Trend Analysis Methodology. Journal of Hydrologic Engineering, ASCE, Vol. 17(9):1042–1046.

- Sen, Z., (2014). Spatial Modeling Principles in Earth Sciences. Springer-Nature: 351 p.  
 Sen, Z., (2017). Innovative trend significance test and applications. Theor. Appl. Climatol. Vol. 127: 939–947.  
 Sen, Z., Mohorji, A.M. and Almazroui, M., (2017). Engineering risk assessment on water structures under climate change effects. Arabian J. Geoscience, Vol. 517, <https://doi.org/10.1007/s12517-017-3275-7>.  
 Thomas, H.A., and Fiering, M.B., (1962). Mathematical synthesis of stream flow sequences for the analysis of river basins by simulation. In: Design of Water Resources Systems.



# Climate Change

7

## 7.1 General

Climate change and variability are the two prime concerns for future sustainable water, food, energy, agriculture, economy, environment, population, and industry and technology development limitations. These issues have been brought into the attention of governments first in 1992 during the Rio de Janeiro summit, where the first international decision was about the atmospheric carbon dioxide ( $\text{CO}_2$ ) amount, which implied temperature increase and rainfall increase or decrease depending on the geographical location.

Anthropogenic forcing due to greenhouse gases, aerosols, land use and changes affect jointly present-day climate change possibility based on scientific studies. Human activities have exerted a net effect due to demographic and economic growth in addition to consumptive lifestyle all of which require energy consumption. The primary energy source is the fossil fuels, and their consumptions emit various gasses into the atmosphere among which carbon dioxide ( $\text{CO}_2$ ) is the most single one that causes global warming. Consequently, there is a steady increase in the temperature records since four decades. Instrumental temperature measurements since 1850 indicate global warming through  $\text{CO}_2$  records, and about  $0.76\text{ }^{\circ}\text{C}$  increment took place during 1850–2005 periods (IPCC 2007; Kundzewitz et al. 2007). Unprecedented global and local changes are taking place in climate structure, and IPCC (2007) report stated that the climate started to warm since the beginning of the industrial revolution from mid-nineteen centuries and it is expected to become more pronounced in the coming years. Global warming as described by IPCC is as unequivocal and very likely (more than 90% certainty) to be related to human activities and so will continue in the future.

Important sectors such as energy, agriculture, industry, transportation, urbanization, waste, vegetation, forestry, land use and land-use change (LULUC), water resources, and biodiversity for sustainable development are subject to future climate change impacts, and therefore, their

vulnerability requires short- and middle-term projections to reduce vulnerability and preparedness for mitigation. Projections on monthly basis provide a dynamic and effective tool for future planning, operation, conservation, and rational management on societal, industrial, sectorial, and stakeholder cooperation for prosperous research and development studies. International integration is also necessary to reduce the greenhouse gas emissions and, hence, to hinder further expansion of the climate change impact, which are directly related to primal (fossil) energy source consumption locally, regionally, and globally. Climate change impact activities can be achieved through a general circulation model (GCM) numerical outputs in combination with regional climate scenario downscaling to local meteorology station records. Downscaling procedures harmonize the local data with different climate change scenario numerical values.

It is the main purpose of this chapter to provide the climate change fundamental models including downscaling, upscaling, and regional dependence features of available meteorological variables among a set of meteorology stations. The basic idea is to downscale the regular mesh climate scenario data at a set of nodes to irregularly scattered meteorology station factual data. For this purpose, available meteorological data must be analyzed in a suitable way for identification of temporal variation of climatic data, especially temperature and precipitation data. To facilitate the practical applications, various MATLAB language written programs are presented in the following sections.

## 7.2 General Circulation Models (GCMs)

The discrepancies between general circulation (climate) model (GCM) predictions and the actual observations may be partly due to the complexity and uncertainty of modeling rather complex atmospheric phenomena behaviors. Climate models in their integro-differential equation forms are non-linear and, therefore, very sensitive to minor changes in parameter values (Trenberth 1993). In the final solutions,

sensitivity is as a chaotic behavior in addition to the stochastic feature of the atmospheric events as another source of uncertainty. In arid regions, the model projections may be even less reliable due to sporadic features in rainfall occurrences. Another approximation in the GCM projections is due to its coarse areal resolution over which only spatial averages of meteorological variables (especially temperature and precipitation) become available.

It is also well known that the lack of local knowledge and failure to incorporate all drivers of the regional climate, currently available regional models (Regional Climate Models (RegCMs) have low predictive powers when used for projections from global to regional climate conditions by means of downscaling methodologies (Wigley et al. 1990, Sen 2009). It is necessary to assess the occurrence frequency and amount of wet and dry periods concerning droughts and floods. Glick (1978) suggested noticing the frequency of record-breaking greater than the previous record. The record-breaking years and their numbers are significant quantities.

Discontinuous meteorological variable predictions, such as rainfall, under the greenhouse warming prevalence are more speculative than a continuous meteorological variable, say temperature. IPCC (2007) suggests that global warming may have the following effects on environment.

- (1) The timing and regional patterns of precipitation will change, and more intense precipitation days are likely,
- (2) GCMs used to predict climate change suggest that a 1.5–4.5 °C rise in global mean temperature would increase global mean precipitation at about 3–15%,
- (3) Although the regional distribution is uncertain, precipitation is expected to increase in higher latitudes, particularly in winter. This conclusion extends to the mid-latitudes according to most GCM results,
- (4) Potential evapotranspiration (ET) as water evaporation from the surface and transpiration from plants rise with air temperature. Consequently, even in areas with precipitation increments, higher ET rates may lead to runoff reduction, which further implies a possible reduction in water supply renewability,
- (5) Precipitation increase may cause to more annual runoff occurrence in the high latitudes. In contrast, some lower latitude basins may experience large reductions in runoff and increase in water shortages, because of evaporation increment coupled with precipitation decrement combination,
- (6) Flood frequencies are likely to increase in many areas, although the amount of increase for any given climate scenario is uncertain and impacts will vary among basins. Floods may become more frequent or less frequent in some areas,

- (7) The frequency and severity of droughts could increase in some areas because of decrease in total rainfall leading to more frequent dry spells and higher ET rates,
- (8) The hydrology of arid and semiarid areas is particularly sensitive to climate variations. In these areas, relatively small changes in temperature and precipitation could result in large percentage changes in runoff, increasing the likelihood and severity of droughts and/or floods,
- (9) Seasonal disruptions might occur in the mountainous area water supplies, if more precipitation falls as rain than snow and if the length of the snow storage season reduces,
- (10) Water quality problems may increase at regions, where there is less flow to dilute contaminants from natural and human sources.

## 7.3 Fundamentals of Climate Modeling

There is no other way except that the globally available climate changes scenario data merge with factual meteorological data through effective numerical modeling software. Both scenario and factual data may have dynamic, statistical, stochastic, and probabilistic ingredients (Chap. 6). Any mathematical model structure toward this aim may have either dynamic principles or equations such as mass, energy, and momentum conservations in addition to gas-state expressions in the forms of integro-differential mathematical equations system-based solutions at regular mesh nodes. Statistical downscaling procedures consider temporal and spatial dependence structure of the meteorological variables. Whether dynamic or statistical downscaling, probability, statistic and uncertainty component play a joint role for acceptable and reliable match between the GCM scenario and ground meteorological data.

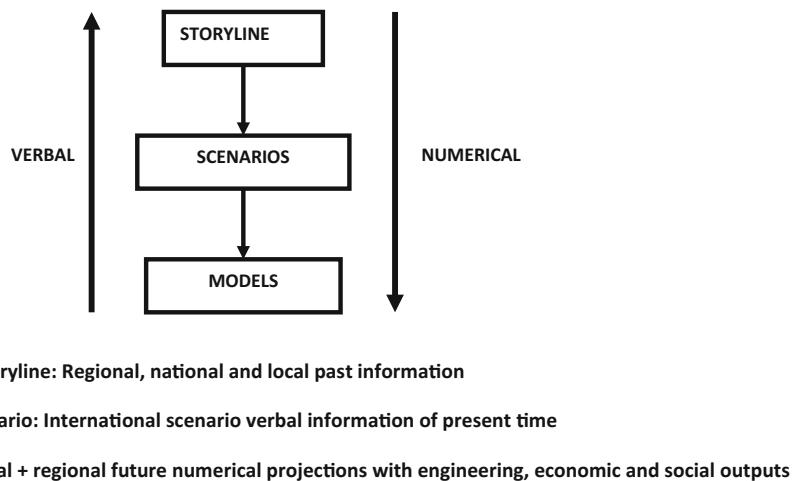
### 7.3.1 Basic Flowchart

Any model should have the basic flowchart as in Fig. 7.1. The flowchart has initially three ingredients as storyline, scenarios, and models, where not only numerical data but also linguistic (verbal) information is employed for successful completion of the preliminary stages (IPCC 2007).

#### 7.3.1.1 Storyline

These are known as past or historical records that are available at each meteorology station starting from the establishment year of the station. They are in the form of time series that represent temporal meteorology variable variations at a fixed location. These series have various

**Fig. 7.1** Model development structure



parametric (mean, mode, median, variance, standard deviation, skewness, kurtosis, etc.) and graphical presentations (histograms, theoretical probability and cumulative distribution functions (PDFs and CDFs) that are useful in a downscaling procedure as will be explained later in this chapter. These parameters and graphs reflect non-serial features of the records. However, as for the serial structure the most important parameter is the correlation coefficient and graphically serial correlation function.

### 7.3.1.2 Scenarios

These are expert view-based projections in the scenario time series forms as the GCM outputs. They are based on expectations concerning energy types, industry, agriculture, environment, population growth, and economy at various stages so that any society or country may choose its future development planning storyline accordingly. These scenario climate change variables are available in the form of monthly or annual numerical series at a set of regularly located nodes over the study area. Hence, these scenario series should be selected according to the node positions and number within the study area including the surrounding adjacent ones.

### 7.3.1.3 Models

The models are useful to match the factual meteorology time series data at a set of irregularly scattered meteorology station locations with the regularly located climate change scenario synthetic time series at a set of nodes. These are referred to as downscaling models, because they carry numerical information from coarsely scattered regular nodes (about 50–100 km apart) to comparatively smaller distance meteorology station set locations. Not only numerical model input and projections, but also verbal (linguistic) information must be considered also in the development and verification

stages of the model. The local verbal information is available from local experts, settlers, or administrators.

The climate change information is available on a global scale as the output of GCMs at very coarse resolutions (about  $250 \times 250 \text{ km}^2$ ). It is necessary to transfer GCM information into finer resolutions. In the climate research studies, the transition from coarser to finer resolution is referred to as “downscaling” (Smith and Tirpak 1989). Since meteorological and climatic measurements are available at a set of ground stations, it is preferable to make downscaling as finer as possible and if possible down to points.

Climate models have restrictions due to a set of assumptions for their manageable operations. For instance, they cannot adequately depict different geographical and spatial variability due to sensitive inferences from atmospheric, oceanic, and land surface-based individual or joint events. Land-use changes such as deforestation and desertification are among the major local climate effective factors, which are not considered in GCM, but they affect local climates substantially. Such regional effects have partial contribution in local meteorological records of temperature, precipitation, and evapotranspiration. Despite restrictive limitations, researchers depend confidently on GCM outputs in their local downscaling works for local projections. One may tend to suspect from the representativeness of current climate model results. There are many alternatives to the GCMs from different centers and accordingly many scenario projections. The statement by Wigley et al. (1990) “in spite of the problems that plague current GCMs, they are the best tools one has for projecting future changes in climate at a regional level” still preserves its validity even today. Nevertheless, the model results “should be treated strictly as scenarios of possible future climate and not as predictions.” This view is also adapted entirely in the work of this chapter.

## 7.4 Downscaling Principles

Present GCM resolution is not enough for social, hydrological, climatologic, and meteorological events (dams, agricultural lands, land use, settlement areas, convective and cyclonic rainfalls, local winds, evaporation, etc.). For instance, clouds occur at sub-grid scales in nature, but their contribution to the GCM must be averaged at coarser scales. The way open for the assessment is to consider either averages over extensive areas, which cause loss of significant information through parameterization, or refined downscaling of GCM node values to local measurement locations with appropriate methodologies. In this book, the second alternative is followed with the hope to catch at least the combined resolution refinement at local scale, where the distance between the measurement stations is less than 30 km in most of the cases. In any GCM, spatial and temporal changes are filed by large grid sizes, and therefore, many local scale impacts, variations, and features are overlooked. At GCM resolution, it is not possible to plan and manage local and environmental activities, because they fall within finer grid size resolution.

The end products of GCMs are available at several centers all over the world. The main problem is to develop a local model, which renders the coarse resolution into a practically finer resolution by taking into consideration the combined effects of GCM outputs with the available ground meteorology measurements such as temperature, precipitation, radiation, humidity, wind speed.

### 7.4.1 Dynamic Downscaling Procedure

Apart from the GCMs, smaller scale regional climate models (RegCMs) are also developed for downscaling purposes using local dynamic ocean-atmosphere-related integro-differential equations, but their downscaling remains at tens of kilometers. Such a downscaling procedure has many finite grid nets, and one of the downscaling sub-areas may or may not include ground stations. In practice, many impact studies require point downscaling values, which are highly sensitive to fine-scale climate change variations. The validity of this statement is true in regions of complex topography such as in mountainous areas, coastal or island locations, and heterogeneous land-cover terrains.

The RegCM software is nested versions of GCMs for dynamic downscaling procedures. Many researchers cannot know what the internal mechanisms of this software are, and therefore, playing with input and output data desired historical data matches with the scenario sequences. Although such regional models are capable to reduce the resolution down to 25–50 km, their internal structure without rational

and logical grasp presents hindrance in practical applications. It is one-way modeling, where the outputs from the GCM enter the RegCM, which implies that the output quality of any RegCM is dependent on the GCM performance. Any restriction and uncertainty in the GCM have exaggeration in the RegCM behavior. Any dynamic modeling procedure poses difficulties, due to the integro-differential equations among which are the following points.

- (a) The differential equations are derived under rather restrictive assumptions of uniformity, homogeneity, isotropy, etc.,
- (b) Geometrical configuration is considered as a set of small 3D cubes with linear changes of the variables inside any cube,
- (c) The final solutions can be applicable theoretically at any time and space. However, practical applications recall the identification of the boundary and initial conditions,
- (d) No element of chance is incorporated in the derivations, and therefore, they describe average behaviors of the system,
- (e) Their solutions are possible only through the numerical techniques (finite element, finite difference, or boundary element). Therefore, rather powerful computers and reliable solution algorithms are necessary,
- (f) Although the equations are expected mechanistically to satisfy the ideal conditions under the light of simplifying restrictive assumptions, their practical solutions are not possible easily and practically. This is because the natural phenomenon such as the climate and meteorological events take place continuously in the atmosphere and their measurements are possible only at a set of irregularly scattered station locations, which do not coincide with the nodal points of finite difference solution mesh.

### 7.4.2 Statistical Downscaling Procedure

Statistical downscaling techniques are potential alternatives to RegCMs, and hence, high spatial and temporal resolution climate scenarios are not required. Although RegCMs are computationally demanding, expensive to run, and difficult to use for long climate simulations, the statistical downscaling methodology requires substantially less computational facilities and produces results that are comparable with RegCMs outputs.

Empirical statistical downscaling is based on the development of mathematical transfer functions or relationships between observed large-scale atmospheric variables at GCM

nodes and the local meteorological variables at a set of irregularly scattered locations within and around the study area. The transfer functions are generally regression based, which are derived between a set of GCMs atmospheric grid node projections and a predictand at the meteorology station.

Environment Agency Rainfall and Weather Impacts Generator (EARWIG) applies “change factors” for a more extensive range of statistics derived from RegCM simulations and is capable of generating stationary climate change scenarios at the point scale or for river catchments in the UK (Kilsby et al. 2007). This approach has been validated against RegCM simulations for the UK Climate Projections (Jones et al. 2009). The use of statistical downscaling requires several assumptions, the most fundamental of which is that the derived relationships between the predictor and predictand will remain valid under conditions of climate change and that the relationships are time-invariant (Yarnal et al. 2001). Von Storch et al. (1993) suggested that if statistical downscaling is to be useful, the relationship between predictor and predictand should explain a large part of the observed variability and that the expected changes in the mean climate should lie within the range of its natural variability, which is generally true for temperature records. Because of the site-specific considerations, the relationship between the large-scale predictors and local outputs often reflects a smaller part of the actual observation variability.

## 7.5 Simple Statistical Downscaling Model

Wilby and Wigley (1997) and Fowler et al. (2007) provided a detailed revision of the statistical downscaling models (SDMs). In practice, there are many SDMs and still many others are bound to appear in the future. SDMs are in continuous improvement to alleviate mismatch by more refined methodologies between the observed and climate scenario output time series.

The SDMs try to establish acceptable empirical relationships between GCM coarse resolution climate output and observed local climate time series. One of the very first SDM is due to Prudhomme et al. (2002), who employed the perturbation method in terms of “change factors” for calculation of the multiplicative or additive difference between the local observations and GCM outputs. Among other SDMs are regression models (Hellström et al. 2001), artificial neural networks (Cavazos and Hewitson 2005), analogue methods based on empirical orthogonal functions (Zorita and von Storch 1999), weather typing schemes

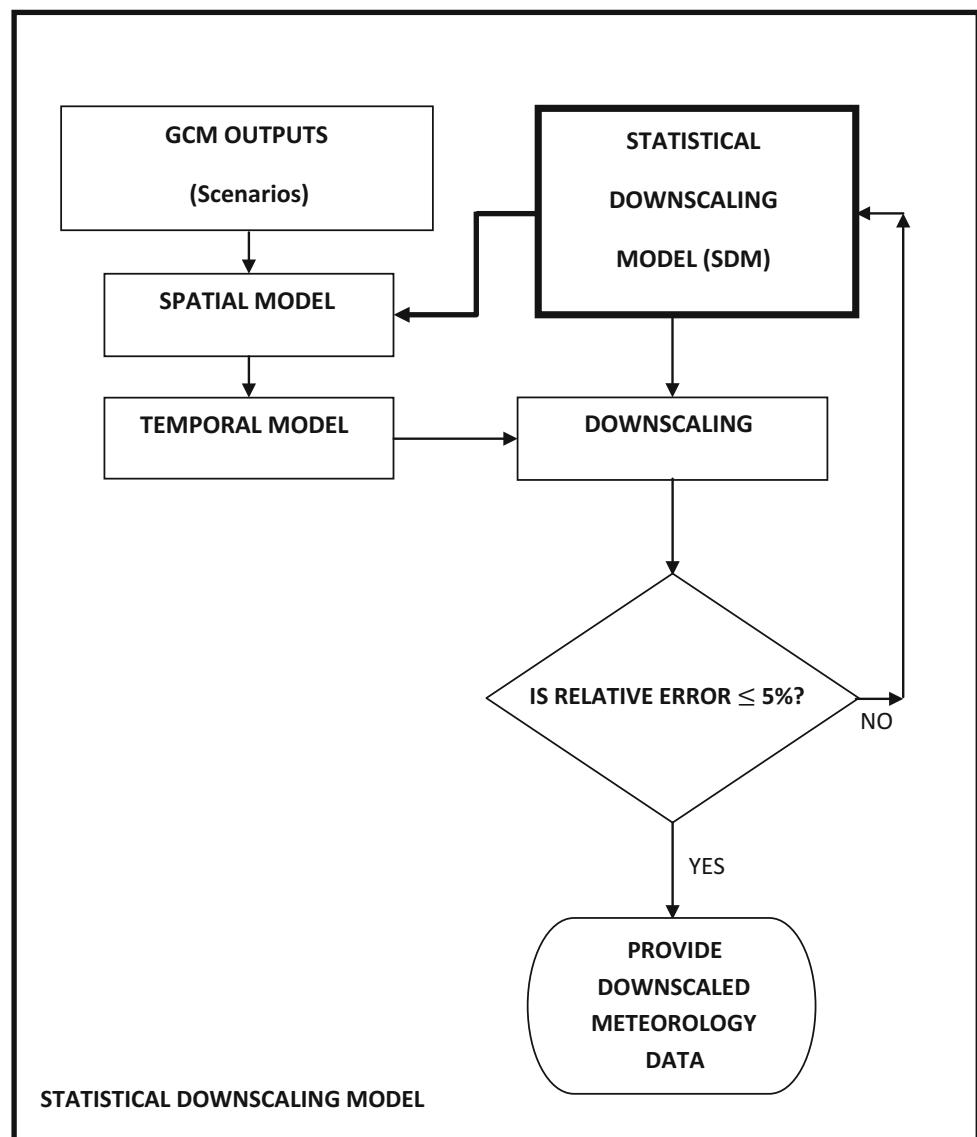
(Goodess and Palutikof 1998), stochastic methods considering weather generators (Wilks 1992; Burton et al. 2010; Willems and Vrac 2011).

The general flowchart of the statistical downscaling method developed by Turkish Water Foundation is given in Figs. 7.2 and 7.3. Its reflection in the literature is presented by Şen (2010), Şen et al. (2010), Dabanlı and Şen (2017). The chart has two branches, international GCM with scenario data and local SDM with any meteorology data primarily temperature and rainfall records. The operation steps of this model are as follows.

- (1) Enter the convenient scenario data from the GCM outputs and factual meteorology data for SDM,
- (2) Combine the two sets of data through the spatial model procedure, which is dependent on the spatial dependence function (SDF), which will be explained in the next section. This procedure transfers the scenario data to meteorology station position,
- (3) Combine the transferred scenario data with the actual meteorology data at each station,
- (4) After the execution of the last two steps statistically the first iteration of the SDM procedure is completed,
- (5) Check the relative error percentage between the final downscaled scenario and meteorology data. If it is not less than 5%, then go back to the spatial model for the next iteration and continue until the desired relative error percentage is satisfied,
- (6) After the completion of SDM procedure, the scenario projections are ready for practical uses,
- (7) In case of drainage basin calculations transfer the projections to another software for drainage basin feature calculations, and if necessary prepare different maps,
- (8) For hydro-meteorological studies on the drainage basin employ any methodology for, say, flood or drought characteristic calculations and, if needed, the associated risks.

This flowchart is convenient for statistical downscaling in any region, and it incorporates a spatial dependence function (SDF) for spatial modeling (Şen 2009) and White Markov (WM) model for temporal modeling (Şen 1974). The model considers all the available rainfall station monthly past records for the construction of SDF for monthly projections up to any desired year such as until 2100. In case of arid regions consideration of four nearby GCM nodes is enough for the SDM. Due to the adaptation of four grid points, the downscaling approach is named as quadratic downscaling (QD) model (Şen et al. 2010, 2012).

**Fig. 7.2** Downscaling model flow diagrams



### 7.5.1 Spatial Dependence Function

Early methods of spatial functions are in the form of hypothetical mathematical expressions depending on the concept that the closer are two stations the higher is the spatial correlation between them. This statement gives the first impression that as the distance increases the regional dependence decreases and after some distance becomes zero. For instance, Gilchrist and Cressman (1954) assumed that the spatial dependence function (SDF) of each station has a parabolic shape. Cressman (1959) has assumed deterministically the radius of influence as 1500, 750, and 500 km to obtain the best regional estimation. Later, Bergthorsson and

Döös (1955) suggested the best interpolation method by parameterization of the Cressman approach. This method is in use in the RegCM dynamic downscaling software. Barnes (1964) has suggested another conceptual SDF like the Gauss curve. Koch et al. (1983) has used the Barnes method in software. A critical view of these methods has been presented by Şen and Habib (2000) and Şen and Öztopal (2001), and an empirical SDF is defined by Şen (2008) as it is used in this paper.

The first step is the calculation of empirical SDF, which reflects the spatial relationship of any station records with others on monthly basis. Calculations of SDFs are obtained through the following MATLAB program.

```

function [RDF,SD]=SpatialDependenceFunction(X,lat,lon)
% This program calculates Regional Dependence Function, RDF, for a given set of
% monthly data in Saudi Arabia
% Written by Zekai Sen on June 2009
% X is the monthly data at a set of stations, IMPORTANT NOTE: There are 27 stations
% SD is the sorted distance in ascending order
% First, calculate the distance sequence
k=0;
for i=1:27
    i1=i+1;
    for j=i1:27
        k=k+1;
        d=100*distance(lat(i),lon(i),lat(j),lon(j));
        Dis(k)=d;
    end
end
[SD,I]=sort(Dis);
% Second, calculate the square difference sequence
for im=1:12
    k=0;
    for j=1:27
        j1=j+1;
        for kk=j1:27
            k=k+1;
            sq=(X(im,j) -X(im,kk))^2;
            SQ(k)=sq;
        end
    end
    % Third, sort SQ in ascending order
    for i=1:k
        j=I(i);
        SSQ(i)=SQ(j);
    end
    % Fourth, calculate cumulative sums
    sum=0;
    for i=1:k
        sum=sum+SSQ(i);
        SSSQ(i)=sum;
    end
    % Fifth, calculate standard sums, subtract from 1 and save the monthly RDF
    % as MRDF
    for i=1:k
        RDF(im,i)=1 -SSSQ(i)/SSSQ(k);
    end
end

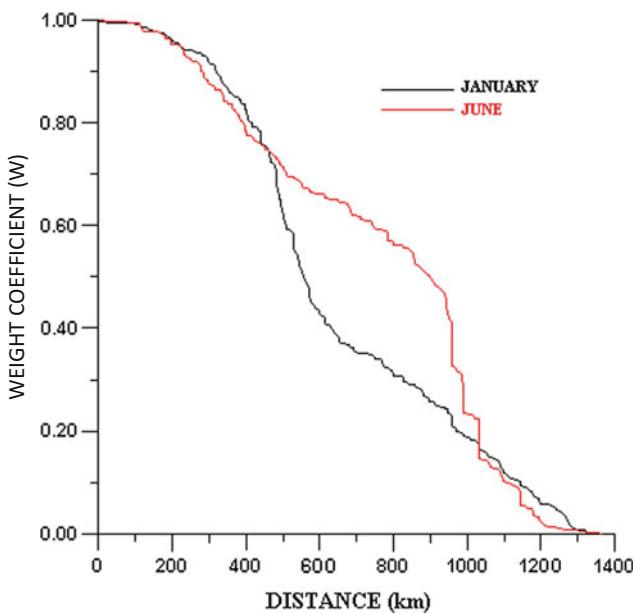
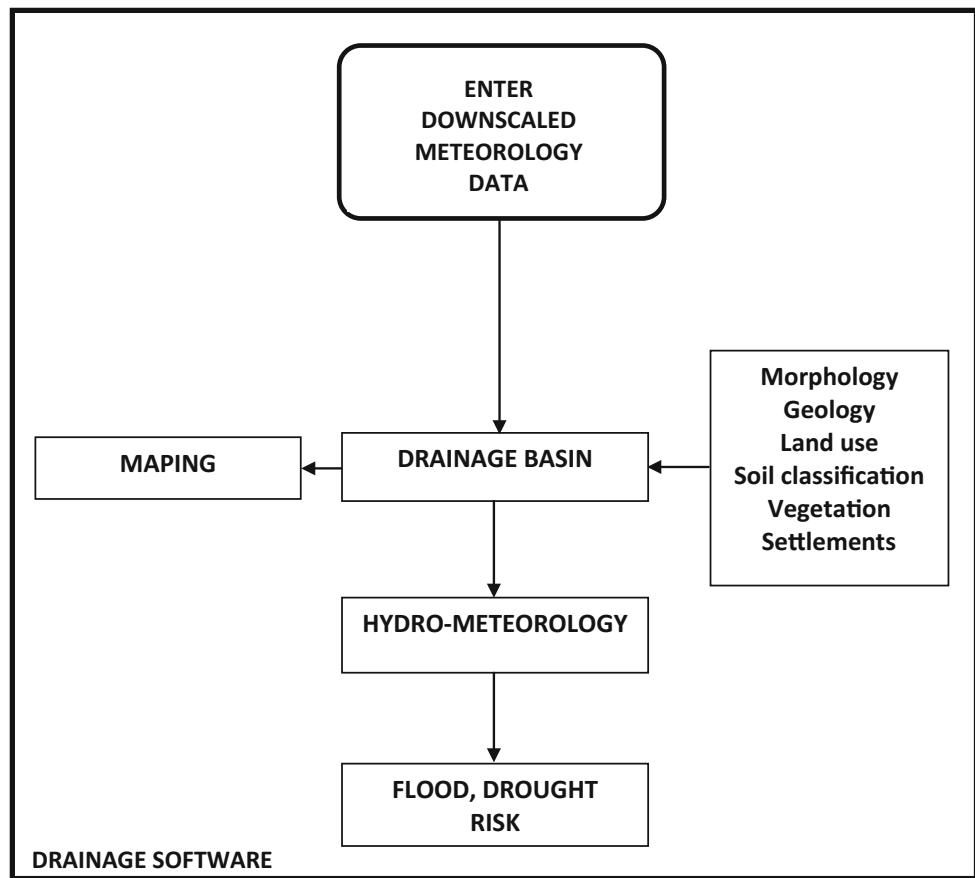
```

As stated above, SDF decreases with the distance increases between the stations. In such a function, all the regional features of data that originate from the combination of morphology (orography), convective cells, North Atlantic Oscillation (NAO) effects, and frontal precipitation occurrences play a role. Figure 7.4 shows a representative SDFs for the months January and June rainfall amounts.

### 7.5.2 Spatial Model

After the identification of the spatial dependence function (SDF), there are different downscaling procedures, where the weighted averages are employed. The weights are obtained from the empirical SDF as in Fig. 7.4. The weighted average expression can be written as,

**Fig. 7.3** Hydro-meteorological calculation chart posterior to SDM



**Fig. 7.4** SDF for Istanbul station

$$Z_p = \sum_{i=1}^n \alpha_i Z_i \quad (7.1)$$

where  $\alpha_i$ 's are the weightings;  $Z_p$  is the predictand;  $Z_i$ 's are predictors. Consideration of percentage property of the weightings, this expression can be written in its most explicit form as,

$$Z_p = \sum_{i=1}^n \left( \frac{W_i}{W_T} \right) Z_i \quad (7.2)$$

where  $W_i$  is the weight attached to  $i$ -th data point and  $W_T$  represents the total weight as the summation of all weights,

$$W_T = W_1 + W_2 + \dots + W_n \quad (7.3)$$

with  $n$  number of data points. The weights are taken by consideration of the station location and month from the SDFs like Fig. 7.4 depending on the distance (on the horizontal axis) between the estimation station and others. The spatial model calculation can be achieved by the following MATLAB program.

```

function [Predict, Dis, Data, err] = SpatialPrediction(D, In, sigma, np)
% This program finds the nearest three points from given data matrix, D,
% with m columns and n rows to a given input data, In, and predicts the
% nn is the number of neighbouring point
% output value, Ou
% The first m -1 columns include the data inputs and the m -th column
% includes data output values
% P is prediction matrix
s=size(D);
n=s(1);
m=s(2);
m1=m-1;
ss=size(In);
ni=ss(1);
mi=ss(2);
% find the distances between the given inputs and data inputs
for k=1:ni
for i=1:n
sum=0;
for j=1:m1
sum=sum+(D(i,j) -In(k,j))^2;
end
dist(i)=sqrt(sum);
end
[Y I]=sort(dist);
for i=1:n
Dis(i,1)=dist(I(i));
Data(i,2)=D(I(i),m);
end
% Prediction for all data values
sum=0;
for i=1:np
sum=sum+gaussmf(Dis(i,1),[sigma 0]);
end
for i=1:np
w(i)=gaussmf(Dis(i,1),[sigma 0])/sum;
end
pre=0;
for i=1:np
pre=pre+w(i)*Data(i,2);
end
err(k)=(pre -Data(k,2))^2;
Predict(k)=pre;
end
end

```

### 7.5.3 Temporal Model

Many stochastic processes are in use to match the observation sequence indistinguishably to the SDM products. The White Markov Process (WMP) is suggested in this chapter for temporal modeling, and it differentiates from conventional stochastic processes on two features as already explained in Chap. 6. It is simple and fast in terms of calculations and also preserves the serial correlation coefficient,  $\rho$ , which is the criteria of the short-term dependency (Sen 1974). It complies with simple, fast, and fractional Gaussian processes by Mandelbrot (1971) to preserve even long-term dependencies.

Basically, WMP consists of an independent process mixture with the first-order Markov process as explained in Chap. 6. The reason why this process is called “white” is due to the normal independent process put onto the classical Markov process. Completely independent process includes the waves in every frequency in the same ratio, and this is called “white light.”

Generally, WMP,  $Z_t$ , a white independent process,  $\eta_t$ , with zero mean and  $\beta$  standard deviation is added to the Markov process,  $X_t$ , as,

$$Z_t = X_t + \beta\eta_t \quad (7.4)$$

The  $k$ -th order dependency coefficient  $\rho_k$  can be calculated in terms of the Markov process (Chap. 6) autocorrelation coefficient,  $\rho_M$ , as,

$$\rho_k = \alpha \rho_M^k \quad (7.5)$$

Here,  $\alpha$  represents the ratio of the standard deviation of white (independent process) and Markov process as,

$$\alpha = 1/(1 + \beta^2/\sigma_M^2) \quad (7.6)$$

With substitution of the Markov process autocorrelation function as,  $\rho_M^{k-1}$ , at  $k$ th lag in the correlation coefficient of the WMP, the previous equation turns into,

$$\rho_k = \alpha \rho_M^{k-1} \quad (7.7)$$

The WMP shows a simple, but effective features of the ARIMA (1, 0, 1) processes, which is frequently used in hydrological studies (Sen 1974). The MATLAB software for the temporal model piece is given in the following software.

## 7.6 Software for Statistical Downscaling

Usage of the computer models became certain scientific requirement relating the past event behavior to future scenario patterns to obtain projections. The major steps in SDM include are the following (Sen 2009).

- (1) Global circulation model (GCM) scenarios: These constitute dynamics base of the future modeling work. Scenario data are calculated via high-capacity computers and technology at one of the major research centers (USA, Canada, UK, Germany, Italy, Japan, and Australia) at a set of grid nodes,
- (2) Special Report on Emissions Scenarios (SRES): These are climate change sub-models, which make additions to GCMs by taking into consideration socio-economic, technology, agriculture, energy variety, and demographic structure of a society (IPCC 2007). These scenarios have been replaced by the Representative Concentration Pathways (RCP) scenarios (Meinshausen et al. 2011),

```
function [X,Y] = WhiteNoiseProcess(Ave,Std,Rho,gam,N)
% This program generates first order Markov process
% Ave = Arithmetic average
% Std = Standard deviation
% Rho = First order correlation coefficient
% Gam = White noise standard deviation
% N = The number of generation data
% X = The synthetic first order Markov time series
% Y = The synthetic white noise time series
eps(1,1:N)=randn(1,N);
% Warm up period
X(1,1)=Ave + Rho*(randn - Ave) + Std*sqrt(1 - Rho*Rho)*randn;
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i - 1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i - 1);
end
%Proper series generation
eps(1,1:N)=randn(1,N);
X(1,1) = X(1,N);
for i=2:N
    X(1,i)= Ave + Rho*(X(1,i - 1) - Ave) + Std*sqrt(1 - Rho*Rho)*eps(1,i - 1);
end
% White noise addition
W=random('normal',0,gam,[1,N]);
for i=1:N
    Y(1,i)=X(1,i)+W(1,i);
end
end
```

- (3) Downscaling model: Any statistical downscaling model transfers GCM outputs to local stations through some statistical methodology. As explained in the previous section, these are SDF [Şen et al. \(2010\)](#) and the WMP by [Şen \(1974\)](#),
- (4) Field models: These models have been generated by means of SDF as developed by [Şen \(2009\)](#).

Scenario-coupled GCM outputs from a convenient center are used as basic global meteorology information (temperature, rainfall, relative humidity, evaporation, wind speed, and sunlight) given at the set of grid points as shown in Fig. 7.5 for Turkey.

Figures 7.2 and 7.3 indicate implicitly the schematic representation of model flow diagram with two basic units as GCM and SDM parts. So, it is possible to downscale the global model results verbally and numerically in the modeling structure. According to flow diagram in Figs. 7.2 and 7.3, initially two parallel modeling branches—GCM and SDM—are coupled through the spatial and temporal sub-models leading to projections at any meteorology station.

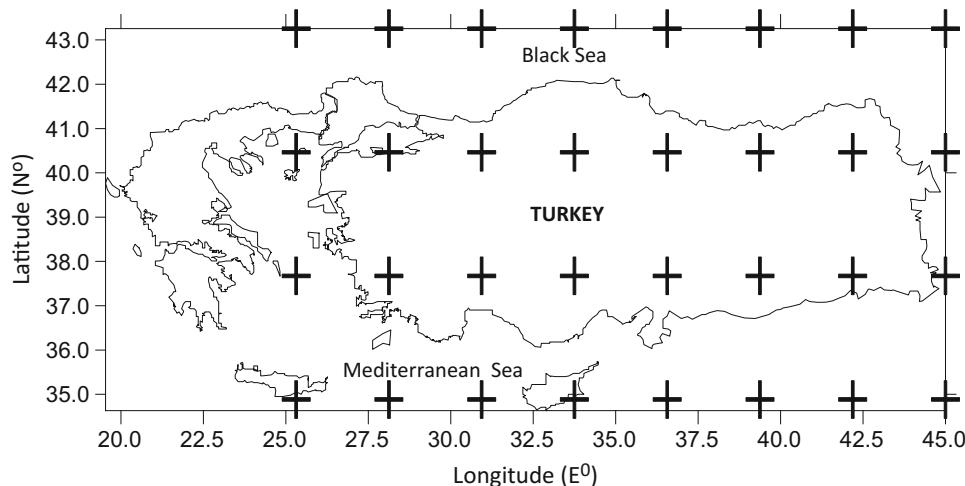
The validation stage includes the comparison of various statistical parameters of the historical and the proposed SDM

output monthly time series. The following steps are important for the model acceptance.

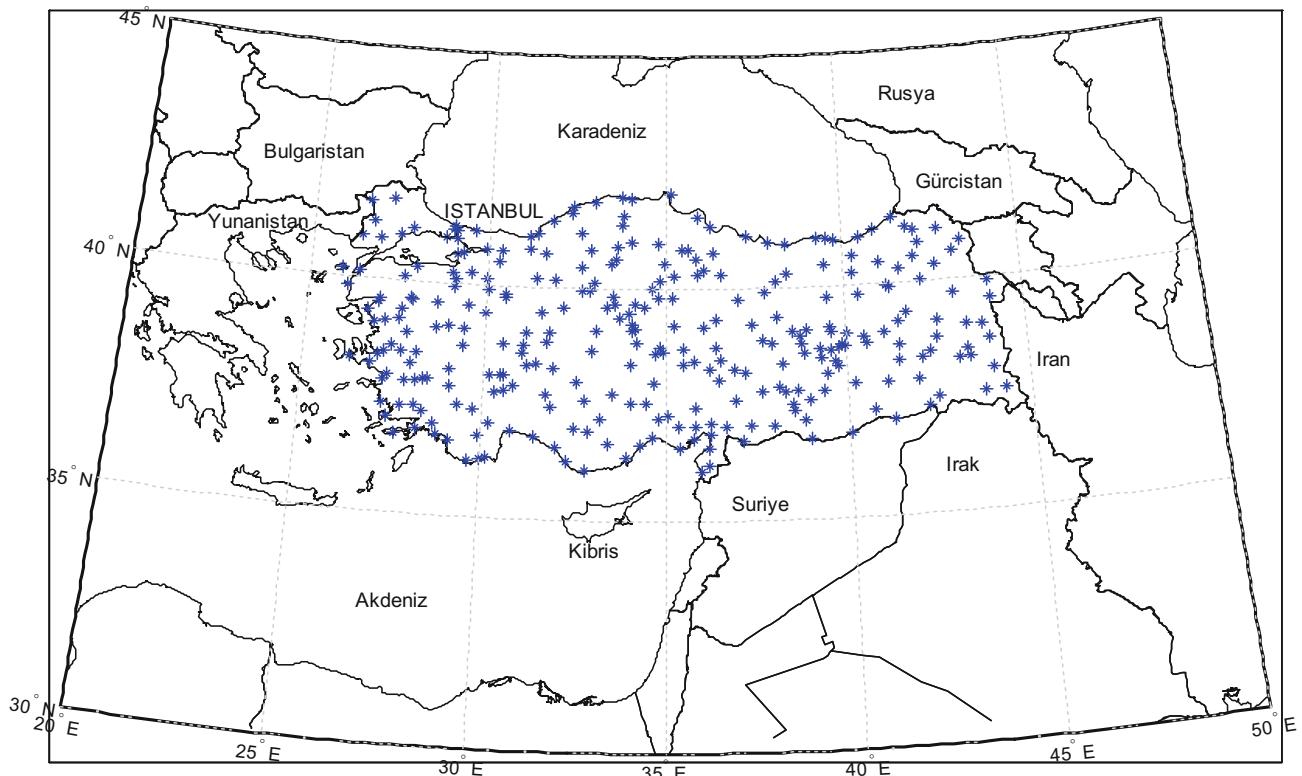
- (1) Adaptation of GCM outputs according to a selected scenario at a set of grid points as indicated in Fig. 7.5. For this work, many scenarios are downloaded as the GCM output,
- (2) Figure 7.6 shows the locations of 311 meteorology stations within Turkey with past meteorology variable records at least for 40 years (480 months). The source of data is the Meteorology General Directorate (Meteorology Genel Merkezi, MGM). The reliability of each meteorology time series is controlled, and finally, 299 stations are considered as reliable for monthly precipitation records.

Comparison of Figs. 7.5 and 7.6 indicates how different the scales of GCM grid point and actual meteorology station locations are. The solution sought is to match spatial resolution between these two maps.

- (3) Temporal model is for different statistical parameters matching between the observation and GCM output monthly time series, where the most important aspect is



**Fig. 7.5** Grid nodes over Turkey



**Fig. 7.6** Locations of meteorology stations in Turkey

to preserve the correlation structure of the observation series with the trend components,

- (4) Finally, the validation of the estimation replicates (combination of GCM outputs with local data) is decided based on the overall relative error that is adopted as  $\pm 10\%$ . If this threshold error level is not satisfied, then temporal pre-whitening iteration with different parameter is continued until the error reduction is brought down to the desired level.

Reduction of the uncertainty in the climate forecasts necessitates that the spatial and temporal nonlinear events in the climate system should be considered in any downscaling modeling. Spatial nonlinearity is concerned with the effects of morphological, meteorological, and hydrological features, and it is accounted by the SDF. Temporal nonlinearity is depicted with the WMP as already explained in Sect. 7.5.3.

The following MATLAB software is developed for the execution of all the steps and procedures explained above.

```

function[ssf, norg, nors]=StatisticalDownscalingProgram(gd,sd)
% This program is written on 4 October 2008 by Zekai SEN, and updated on
% 14 April 2015.
% It downscales the GCM scenario data to a given meteorology station data
% by means of Spatial Dependence Function
% gd = Observation data sequence
% sd = Scenario data sequence
% ss = Last scenario data (adjusted sequence)
% ssf = Regionally adjusted projection data on arithmetic means,
% standard deviations and serial correlation coefficients
% ssssf = Apart from the serial correlation monthly arithmetic average and
% standard deviation adjustment data sequence
% Observation and scenario data monthly arithmetic average and standard
% calculations
% IMPORTANT NOTE! GAMMA PDF DATA VALUES CANNOT BE ZERO AND THE
% AT LINE 94. ADJUSTMENT MUST BE VERIFIED FOR EACH STATION ACCORDING TO
% For "Correlation" function run, it must be in the current folder.
ng=max(size(gd));
ns=max(size(sd));
for i=1:12
    meang(i)= mean(gd(i:12:ng));
    stdg(i)= std(gd(i:12:ng));
    means(i)= mean(sd(i:12:ns));
    stds(i) = std(sd(i:12:ns));
end
% Observation and scenario data annual durations
ngy=ng/12;
nsy=ns/12;
% Observation and scenario data empirical distribution
ngl=ng+1;
for i=1:ng
    pgd(i)=i/ngl;
end
ns1=ns+1;
for i=1:ns
    psd(i)=i/ns1;
end
% Sorting of observation and scenario data in ascending order
dizgd=sort(gd);
dizsd=sort(sd);
% Observation and scenario data Gamma distribution parameters
gampg=gamfit(gd);
gamps=gamfit(sd);
% Variability domain of observation and scenario data
gddeg=min(gd):0.1:max(gd);
sddeg=min(sd):0.1:max(sd);
% Observation data empirical and theoretical graphs
figure(1);
gdgam=qamcdf(gddeg,gampg(1,1),gampg(1,2));
plot(dizgd,pdg,'gddeg',gdgam,'LineWidth3');
legend('Calculation','Theory','Location','SouthEast');
xlabel('Monthly values (mm) [OBSERVATION]');
ylabel('Nonexceedence probability');
text(10,0.95,'GAMMA PDF');
text(10,0.9,[alpha] = num2str(gampg(1)));
text(10,0.85,[beta] = num2str(gampg(2)));

figure(2);
sdgam=qamcdf(sddeg,gamps(1,1),gamps(1,2));
plot(dizsd,psd,'sddeg',sdgam,'LineWidth3');
legend('Calculation','Theory','Location','SouthEast');
xlabel('Monthly value (mm) [SCENARIO AT POINT]');
ylabel('Nonexceedence probability');
text(10,0.95,'GAMMA PDF');
text(10,0.9,[alpha] = num2str(gamps(1)));
text(10,0.85,[beta] = num2str(gamps(2)));

% Observation and scenario data Gamma disprobabilities
gampgd=gamcdf(gd,gampg(1,1),gampg(1,2));
gampsd=gamcdf(sd,gamps(1,1),gamps(1,2));
% Observation and scenario data standardized Gauss distribution values
norgd=norminv(gampgd,0,1);
norsd=norminv(gampsd,0,1);
for i=1:12
    meannorgd(i)= mean(gd(i:12:ng));
    stdnorgd(i)= std(norgd(i:12:ng));
    meannorsd(i)= mean(norsd(i:12:ns));
    stdnorsd(i) = std(norsd(i:12:ns));
end

```

```
% Standardization of observation data
k=0;
for i=1:ngy
    for j=1:12
        k=k+1;
        norg(k)=(norgd(:,j))/stdnorgd(j);
    end
end
% Standardization of scenario data
k=0;
for i=1:nsy
    for j=1:12
        k=k+1;
        nors(k)=(norsd(:,j))/stdnorsd(j);
    end
end
[kg]=correlation(norg,24);%correlation function works
[ks]=correlation(nors,24);

kk=0:1:10; % Generate 100 kk numbers
kkl=length(kk); %Determine kk sequence length
for i=1:kkl
    for j=1:10
        r=random('normal',0,1,[ns,1]); %Generate ns number of normal (Gaussian) random number
        norsn=nors'+kk(i).*r;%Generates normalized scenario data according to random numbers
        [c1]=correlation(norsn,24);
        c(j,1:24)=c1;
    end
    ca(i,1:24)=mean(c);
end
for i=1:kkl
    hata(i)=sum((kg(1:24,i,1:24)).^2); %Minimum difference between normalized
    % scenario data and the sequence for
    % which the correlation structure is adjusted.
end
[hmin,I]=min(hata); %Find the minimum error and its line (row)
ceuy=c(I,1:24);
figure(3) %ADJUST CORRELATIONS
plot(1:24,ceuy,'-',1:24,kg,'k:', 'LineWidth',2);
legend('Most suitable correlation',...
 xlabel('Lag');
 ylabel('Serial correlation'); % Due to generation of random numbers one cannot
 % expect complete match in Figure 3.
figure(4)
plot(1:24,kd,'p-',1:24,ks,'k:',1:24,ceuy,'r-','LineWidth',2);
legend('Observation' 'Scenario at point' 'Adjusted scenario')
 xlabel('Lag');
 ylabel('Correlation coefficient');

% MATCH AND FIND THE CORRESPONDANCE OF NORMAL DISTRIBUTION PROBABILITIES OF FINAL STANDARD
SCENARIO DATA
% (norsn- here the last letter is used for 'new' meaning) TO OBSERVATION DATA
% Normal distribution probabilities and their corresponding Gamma value
% Herein, sss implies scenario data without any adjustment in serial correlation structure
normps=normcdf(norsd,mean(norsn),std(norsn));
sss=gaminv(normps,gampg(1,1),gampg(1,2));
normp=normcdf(norsn,mean(norsn),std(norsn));
ss=gaminv(normp,gampg(1,1),gampg(1,2));
% Monthly parameters of generated scenario data
for i=1:12
    meangss(i)= mean(ss(i:12:ns));
    stdgss(i)= std(ss(i:12:ns));
    meangsss(i)= mean(sss(i:12:ns));
    stdgsss(i)= std(sss(i:12:ns));
end
% Standardization of scenario data
k=0;
for i=1:nsy
    for j=1:12
        k=k+1;
        sgam(k)=(ss(:,j))/stdgss(j);
        sgams(k)=(sss(:,j))/stdgsss(j);
    end
end
% Adjustment of monthly parameters
% Here sss implies scenario data without serial correlation
% average and standard deviation adjustments
```

.....Continued.....

```

k=0;
for i=1:nsy
    for j=1:12
        k=k+1;
        ssf(k)=sgam(k)*stdg(j)+meang(j);
        sssf(k)=sgams(k)*stdg(j)+meang(j);
    end
end
% Limitation of negative data values
ssf(ssf<0)=0;
sssf(sssf<0)=0;
% Observation data histogram
figure (5)
hist(gd,15);
xlabel('Monthly values (mm) [OBSERVATION]');
ylabel('Frequency');
% Histogram of the finally adjusted scenario data
figure(6)
hist(ssf,15);
xlabel('Monthly values (mm) [ADJUSTED SCENARIO]');
ylabel('Frequency');
% Serial correlation adjusted scenario data
figure(7)
hist(sssf,15);
xlabel('Monthly values (mm) [SCENARIO AT POINT]');
ylabel('Frequency');
% Generated scenario data monthly arithmetic averages
for i=1:12
    meansss(i)= mean(ssf(i:12:ng));
    stdss(i)= std(ssf(i:12:ng));
    meanssss(i)= mean(sssf(i:12:ng));
    stdsss(i)= std(sssf(i:12:ng));
end
% Comparison graph of observation and generated scenario data monthly arithmetic averages.
figure(8)
plot(1:12,meang,'b-',1:12,meansss,'r--',1:12,meanssss,'k:','LineWidth',2);
legend('Observation','Adjusted scenario','Scenario at point','Location','Southeast');
xlabel('Months (Jan., Feb., ..., Dec.)');
ylabel('Monthly average rainfall (mm)');
% Comparison graph of observation and generated scenario data monthly standard deviations.
figure(9);
plot(1:12,stdg,'b-',1:12,stdss,'r--',1:12,stdsss,'k:','LineWidth',2);
legend('Observation','Adjusted scenario','Scenario at point','Location','Southeast');
xlabel('Months (January, February, ..., December)');
ylabel('Monthly standard deviations (mm)');
% Comparison graph of observation and generated scenario data monthly correlation coefficients
figure(10);
[cg]=correlation(gd,24);
[css]=correlation(ssf,24);
[csss]=correlation(sssf,24);
plot(1:24,cg,'b-',1:24,css,'r--',1:24,csss,'k:','LineWidth',2);
legend('Observation','Adjusted scenario','Scenario at point');
xlabel('Lag');
ylabel('Correlation coefficient');
end

```

The “correlation” coefficient MATLAB program is given in the following box, because it is necessary for the “StatisticalDownscalingProgram” run.

```

function [cor]=correlation(x,ml)
% This program is written in 1975 by Zekai Sen
% x is the time series vector
% ml is the maximum lag
n=max(size(x));
s=std(x);
m=mean(x);
for i=1:n
    y(i)=(x(i)-m)/s;
end
for i=1:ml
sum=0;
ne=n-i;
for j=1:ne
sum=sum+y(j)*y(i+j);
end
cor(i)=sum/ne;
end

```

In the following box quadratic downscaling MATLAB program is presented for arid regions and also for preliminary simple downscaling calculations (Sen 2012).

```

function [StationScenario]=QuadrapleDownScalingSA(ScenarioVar,GridLatLon,lat,lon,StationNo)
% This program is written by Zekai Sen on December 2009 Sunday for PRINCE
% SULTAN CENTER
% ScenarioVar = A two -dimensional matrix with set of grid points NG and
% monthly rainfall amounts NM
% GridLatLon = The matrix corresponding to the locations of the grid
% points in terms of longitudes and latitudes: Should come from the main
% program as "ArabScenarioLatitudeLongitude"
% lat = Station latitude: Should come from the main program as "latARAB"
% lon = Station longitude: Should come from the main program as "lonARAB"
% StationScenario = This is the scenario sequence of the meteorological
% variable as SCanarioVar
% StationNo = This is the number of stations as given by the author. For
% instance, Riyadh station number is 14 in the sequence
% First, calculate distances between a given meteorology station and grids;
[NM NG]= size(ScenarioVar);
k=0;
for i=1:NG
    k=k+1;
    d=100*distance(lat(StationNo),lon(StationNo),GridLatLon(i,1),GridLatLon(i,2));
    Dis(k)=d;
end
[SD,I]=sort(Dis);
% Estimate the monthly unadjusted rainfall value from 4 nearby grids
% scenario according to inverse distance square technique
for im=1:12
    for km=im:12:NM
        sm1=0;
        sm2=0;
        for ig=1:4
            it=I(ig);
            sm1=sm1+ScenarioVar(km,it)*(1/SD(ig))^2;
            sm2=sm2+(1/SD(ig))^2;
        end
        StationScenario(km)=sm1/sm2;
    end
end

```

On the other hand, the following MATLAB program provides opportunity to match between the observation and scenario data.

applications. Herein, only 10 stations are considered for the application of the downscaling methodology (Table 7.1).

In Fig. 7.7, observations and downscaled rainfall value PDFs are presented for comparison purposes.

```

function [AII] = ObservationScenarioSeriesMatch(O,S,F)
% This program statistically match es observation and climate change series
% G      : Monthly observation series (Yearx12)
% S      : Monthly scenario series (Yearx12)
% F      : Future climate change data
% AII    : Arithmetic average and standard deviation adjusted data
n=length(O(:,1));
go=mean(O); % Monthly arithmetic average of observation data
gs=std(O); % Monthly standard deviation of observation data
io=mean(S); % Monthly arithmetic average of scenario data
is=std(S); % Monthly standard deviation of scenario data
% Correlation coefficients between successive monthly observation series
for i=1:11
    cc=corrcoef(O(:,i),O(:,i+1));
    gc(i)=cc(1,2);
end
cc=corrcoef(O(:,12),O(:,1));
gc(12)=cc(1,2);
% Correlation coefficients between successive monthly scenario series
for i=1:11
    cc=corrcoef(S(:,i),S(:,i+1));
    ic(i)=cc(1,2);
end
cc=corrcoef(I(:,12),S(:,1));
ic(12)=cc(1,2);
% Standardization of scenario data
for i=1:12
    for j=1:n
        IS(j,i)=(S(j,i) - io(i))/is(i);
    end
end
% Adjust observation and scenario series with the same arithmetic average and
% standard deviation and conserve the result as first approximation matrix AII
for i=1:12
    for j=1:n
        AII(j,i)=IS(j,i)*gs(i)+go(i); % First adjustment stage
    end
end
% The second adjustment should be achieved based on correlation
% coefficients
oA=mean(AII);
sA=std(AII);
% END OF STATISTICAL SIMILARITY
%
% ADJUSTMENT OF FUTURE PROJECTION (2000 -2050)
m =length(F(:,1)); % YEAR LENGTH OF THE FUTURE CLIMATE CHANGE DATA
end

```

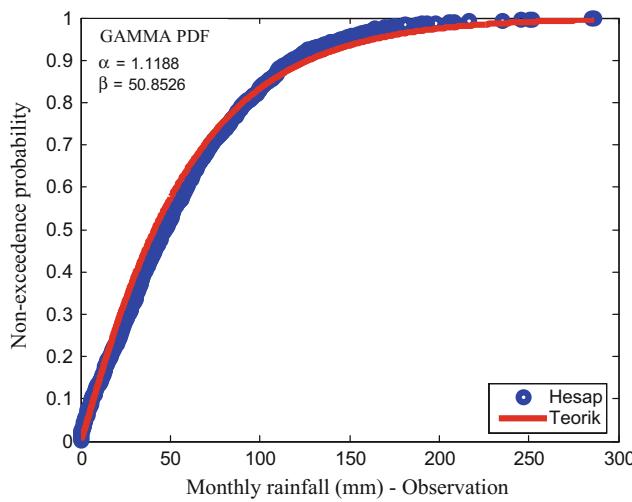
### 7.6.1 Application in Turkey

The GCM output data are downloaded from Max Plank Institute at nodes in Fig. 7.5, and they have about 250–300 m resolution, which is not practically applicable in many studies, and therefore, they are downscaled to the meteorology station locations in Fig. 7.6 by spatial and temporal model

It is observed that empirical rainfall PDF confirms with theoretical Gamma PDF of given  $\alpha$  and  $\beta$  parameters. This conformity provides various calculation possibilities for future projections. The empirical frequency distribution diagrams are given for observation and theoretical Gamma distribution projections in Fig. 7.8.

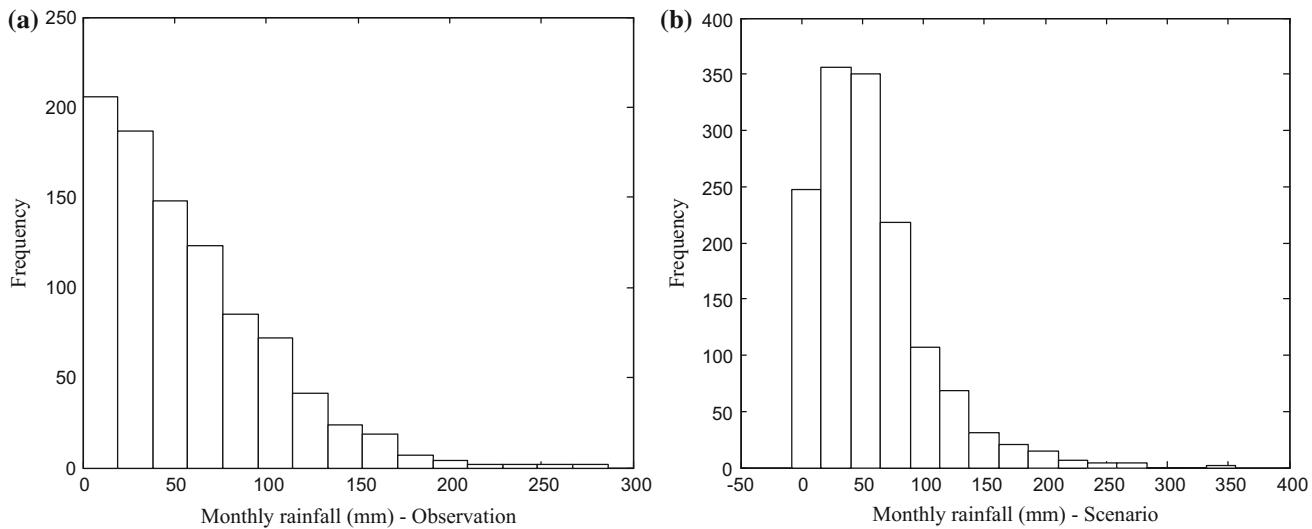
**Table 7.1** Station features

Station name	Station number	Latitude (E)	Longitude (N)	Annual precipitation	
				Mean (mm)	St. Dev. (mm)
Bursa	42	29,00	40,22	691	145
Edirne	12	26,55	41,68	591	135
Istanbul	18	29,08	40,97	680	117
Izmit	19	29,93	40,77	777	151
Karamürsel	44	29,28	40,67	733	133
Bartın	2	30,55	41,18	1218	246
Biga	29	26,40	40,13	698	142
Enez	127	26,15	41,68	694	152
Kırklareli	13	27,22	41,73	564	134
Ereğli	41	27,21	41,65	698	142

**Fig. 7.7** Frequency diagrams of the monthly rainfall data: a observation, b scenario data

This figure implies that the frequency diagrams resemble each other in terms of degree, and after comparison one can reach to the following points.

- (1) In general, the PDF of the observation data has exponential (low rainfalls appear more frequently than others), and hence, it is understood that the rainfall distribution estimation confirms with the Gamma PDF (Fig. 7.7),
- (2) When the observation and theoretical PDFs are compared, one can see that drought rainfall in the scenario data is expected to increase in the coming years; most of the dry seasons will not be the most frequent one,
- (3) While it is expected that high rainfall events may occur more frequently (200 mm and more), very extreme rainfall has never seen before (250 mm and more),
- (4) According to the extractions from the graphs that cover whole Turkey, it is estimated that the rainfall in some stations (such as in İstanbul) may not decrease

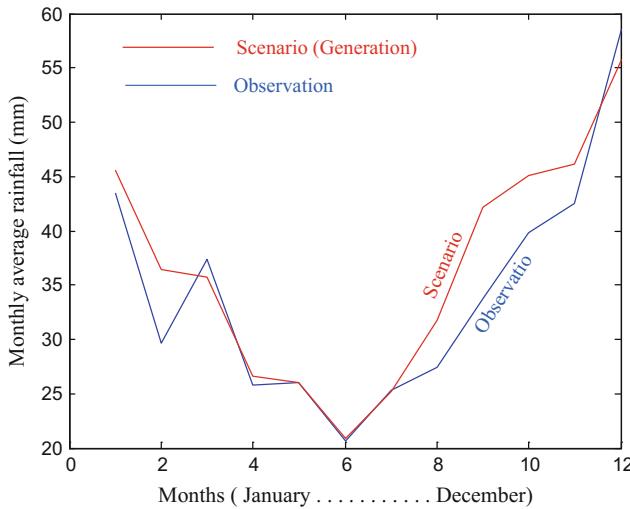
**Fig. 7.8** Frequency distribution functions of theoretical and calculated rainfall

dramatically, but the frequency of the drought and extreme rainfall may increase. Such results may be forecasted reliably and in more detail for all the nodes.

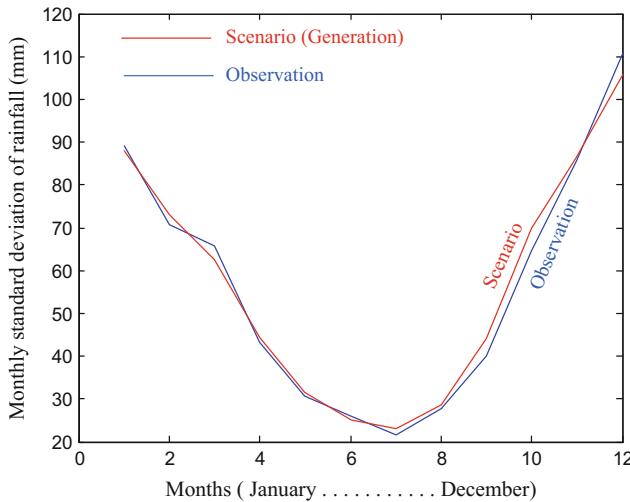
In Fig. 7.9, monthly averages of observation and estimation series are presented, and the first glance indicates that they are in good agreement.

The observation and scenario monthly average rainfalls resemble each other, but in winter season there is not a significant difference, while in the summer season monthly average estimations appear more frequently.

Figure 7.10 shows the monthly average standard deviation values from the observation and estimation data. The standard deviation can also be named as “rainfall diversity,”



**Fig. 7.9** Observation and estimation monthly average rainfalls



**Fig. 7.10** Observation and scenario monthly average rainfall standard deviations

and they follow each other within practically acceptable limits closely.

At this stage, the proposed SDM provides estimations that are in good harmony also in terms of standard deviations.

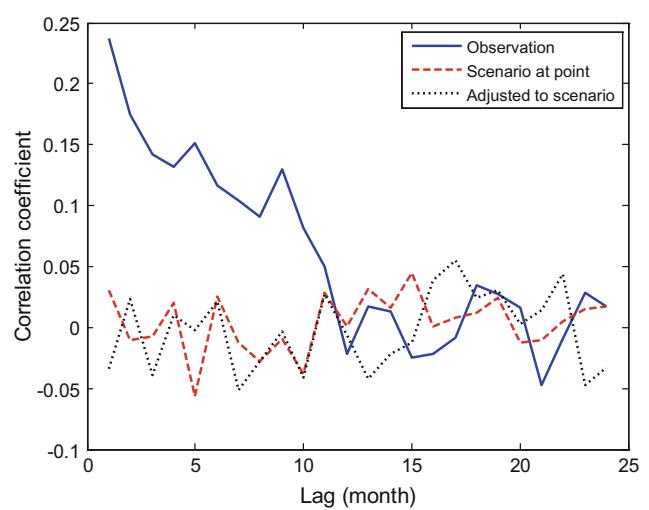
The most important time-wise feature of the SDM is to satisfy serial correlation coefficient indistinguishability between observation and projection data. This is the most important step in downscaling works in any climate modeling study. If the given observation time series is  $X_i$ , then the standard series  $x_i$  is obtained as follows,

$$x_i = \frac{X_i - \bar{X}}{S_X} \quad (7.8)$$

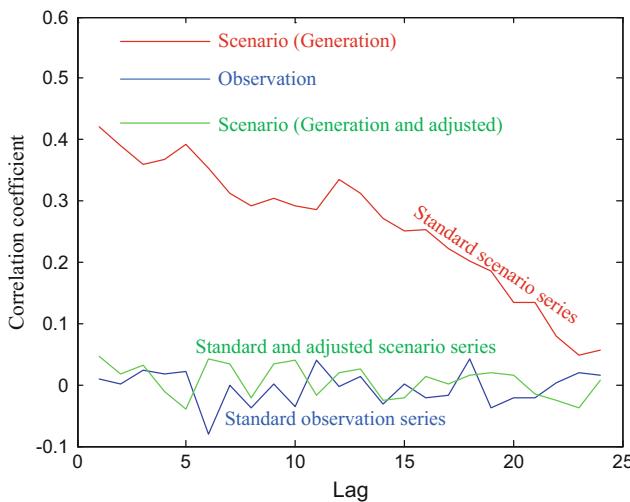
where  $\bar{X}$  is the average of the original time series and  $S_X$  is the standard deviation. Monthly correlation coefficient changes are shown in Fig. 7.11, and it is obvious that there is a good agreement between the two series.

To give an idea about the downscaling concerning the serial correlation function, Fig. 7.12 is given for relevant discussions. The serial correlation function is adjusted based on standard series, because averages and standard deviations are already preserved in the model. Serial correlation adjustment cannot be tackled easily, and it needs trial-and-error method in a systematic way by adjusting  $\alpha$  parameter in Eq. (7.6), and this is satisfied by the MATLAB software. As obvious from Fig. 7.12, spatially scenario downscaling series correlation function is adjusted in such a way that it has indistinguishable pattern from the temporal (WMP) adjustment series.

After all the above-performed frequency distribution, monthly average, standard deviation, and serial correlation



**Fig. 7.11** Observation and estimation serial correlation functions



**Fig. 7.12** Serial correlation adjustments

coefficient settings, projection data for different stations can be achieved by the same statistical characteristics with the observations.

## References

- Barnes, S. L., 1964: A technique for maximizing details in numerical map analysis. *Journal of Applied Meteorology*, Vol. 3: 395–409.
- Bergþorsson P. I., and Döös, B. R., (1955). Numerical Weather Map Analysis. *Tellus*, Vol. 7, No. 3, Stockholm, Sweden, Aug.: 329–340.
- Burton, A., Fowler, H. J., Blenkinsop, S., and Kilsby, G. S. (2010). Downscaling transient climate change using a Neyman–Scott Rectangular Pulses stochastic rainfall model. *Journal of Hydrology*, Vol. 381: 18–32.
- Cavazos, T., and Hewitson, B.C., (2005). Performance of NCEP–NCAR reanalysis variables in statistical downscaling of daily precipitation. *Climate Research*, Vol. 28: 95–107.
- Cressman, G. P., (1959). An Operational Objective Analysis System. *Monthly Weather Review*, Vol. 87, No. 10, Oct.: 367–374.
- Dabani, I., and Şen, Z., (2017). Turkish Water Foundation (TWF) statistical climate downscaling model procedures and temperature projections. *European Water*, Vol. 59: 25–32.
- Fowler, H. J., Blenkinsop, S., and Tebaldi, C., (2007). Linking climate change modelling to impacts studies?: recent advances in downscaling techniques for hydrological modelling, *Int. J. Climatol.*, 27, 1547–1578, <https://doi.org/10.1002/joc1556>.
- Gilchrist, B., and Cressman, G.P., (1954). An Experiment in Objective Analysis. *Tellus*, Vol. 6, No. 4, Stockholm, Sweden: 309–318.
- Glick N (1978) ‘Breaking Records and Breaking Boards’, American Mathematical Society.
- Goodess, C.M., and Palutikof, J., (1998). Development of daily rainfall scenarios for southeast Spain using a circulation-type approach to downscaling. *International Journal of Climatology*, Vol. 18: 1051–1083.
- Hellström, C., Chen, D., Achberger, C., and Räisänen, J., (2001). Comparison of climate change scenarios for Sweden based on statistical and dynamical downscaling of monthly precipitation. *Climate Research*, Vol. 19: 45–55.
- IPCC (2007), “Working group II contribution to the intergovernmental panel on climate change fourth assessment report climate change 2007: Climate change impacts, adaptation and vulnerability”: 9–10.
- Jones, P.G., Thornton, P.K., and Heinke,J., (2009). Generating characteristic daily weather data using downscaled climate model data from the IPCC’s Fourth Assessment: 19 pp.
- Kilsby, C.G., Jones, P.D., Burton, A., Ford, A.C., Fowler, H.J., Harpham, C., James, P., Smith, A., Wilby, R.L., (2007). A daily weather generator for use in climate change studies. *Environmental Modeling and Software*, Vol. 22: 1705–1719.
- Koch, S.E., desJardins, M., and Kocin, P.J., (1983). An interactive Barnes objective map analysis scheme for use with satellite and conventional data. *Journal of Climatology and Applied Meteorology*, Vol. 22: 1487–1503.
- Kundzewitz, Z.W., Mata, L.J., Arnell, N.W., Doell, P., Kabat, P., Jimenez, B., Miller, K.A., Oki, T. and Şen, Z. (2007) ‘Freshwater resources and their management’, in Parry, M.L., Canziani, O.F., Palutikof, J.P., Van der Linden, P.L. and Hanson, C.E. (Eds.): *Climate Change 2007: Impacts, Adaptations and Vulnerability, Contribution of Working Group II to the Fourth Assessment Report and of the Intergovernmental Panel on Climate Change*, Cambridge University Press.
- Mandelbrot, B.B., (1971). A fast fractional Gaussian noise generator. *Water Resources Research*, Vol. 7(3): 543–553.
- Meinshausen, M., Smith, S.J., Calvin, K. et al. (2011). *Climatic Change*, 109: 213. <https://doi.org/10.1007/s10584-011-0156-z>.
- Prudhomme, C., Reynard, N., Crooks, S., (2002). Downscaling of global climate models for flood frequency analysis: where are we now? *Hydrological Processes*, Vol. 16: 1137–1150.
- Smith, JB, Tirpak DA (eds.) (1989). *The Potential Effects of Global Climate Change on the United States*. Report to Congress, United States Environmental Protection Agency, EPA-230-05-89-050, Washington, DC, 409.
- Şen Z (1974) Small sample properties of the stochastic processes and the Hurst phenomenon in hydrology. Unpublished Ph. D. Thesis, University of London, Imperial College of Science and Technology, 296 pp.
- Şen, Z., (2008). *Spatial Modeling Principles in Earth Sciences*. Springer Verlag, 351 page.
- Şen, Z., (2009). Precipitation downscaling in climate modelling using a spatial dependence function. *International Journal of Global Warming*, Vol. 1(1/2/3): 29–42.
- Şen, Z., (2010). Critical Assessment of Downscaling Procedures in Climate Change Impact Models, *Journal of Ocean and Climate: Science, Technology and Impacts*, Vol. 1(2): 85–98.
- Şen, Z., (2012). Innovative Trend Analysis Methodology. *Journal Hydrol. Engrg.*, ASCE, Vol. 17(9):1042–1046.
- Şen, Z., and Habib, Z., (2000). Spatial analysis of monthly precipitation in Turkey. *Theor. Appl. Clim.* Vol. 61, 1–16.
- Şen, Z., Öztopal, A., (2001). Assessment of regional air pollution variability in Istanbul. *Environmetrics*, Vol. 12(5): 401–420.
- Şen, Z., Al Alsheikh, A., Alamoud, A. S. A. M., and Abu-Risheh, A. W., (2012). Quadrangle Downscaling of Global Climate Models and Application to Riyadh, *Journal of Irrigation and Drainage Engineering*, Vol. 138(10):918.
- Şen, Z., Uyumaz, A., Cebeci, M., Öztopal, A., Küçükmehmetoğlu, M., Özger, M., Erdik, T., Sırdaş, S., Şahin, A.D., Geymen, A., Oğuz, S. and Karsavran, Y. (2010) The impacts of Climate Change on Istanbul and Turkey Water Resources, 1500p, Istanbul Metropolitan Municipality, Istanbul Water and Sewerage Administration, Istanbul.

- Trenberth, K. (ed.), (1993). Climate System Modeling. Cambridge University Press, Cambridge, UK: 818 pp.
- von Storch, H., Zorita, E. and Cubash, U. 1993: Downscaling of global climate change estimates to regional scales: an application to Iberian rainfall in wintertime. *Journal of Climate* 6, 1161–71.
- Wigley, T.M.L., Jones, P.D., Briffa, K.R. and Smith, G., (1990). Obtaining sub-grid scale information from coarse resolution general circulation model output. *Journal of Geophysical Research*, Vol. 95: 1943–53.
- Wilby, R. L., and Wigley, T. M. L., (1997). Downscaling general circulation model output: A review of methods and limitations, *Prog. Phys. Geogr.*, Vol. 21: 530–548.
- Wilks, D.S., (1992). Adapting stochastic weather generation algorithms for climate change studies. *Climatic Change*, Vol. 22: 67–84.
- Willems, P., and Vrac, M., (2011). Statistical precipitation downscaling for small-scale hydrological impact investigations of climate change. *Journal of Hydrology*, Vol. 402: 193–205.
- Yarnal B, Comrie AC, Frakes B, Brown DP (2001) Review. Developments and prospects in synoptic climatology. *Int J Climatol* 21:1923–1950.
- Zorita, E., and von Storch, H., (1999). The analog method as a simple statistical downscaling technique: comparison with more complicated methods. *Journal of Climate*, Vol. 12 (8): 2474–2489.

# Spatial Analyses

## 8.1 General

Many earth systems phenomena take place over a region continuously by time; hence, the variation is of the spatio-temporal type. The spatial evolvement of the events can be sampled generally at a set of irregular sites or particularly at regular mesh grid nodes as already explained in the previous chapter. One can have temporal measurements at a single site or at a set of single sites either instantaneously or gradually as spatio-temporal data form. Local or regional variations' spatial description, interpretation, and identification can be represented in two- or three-dimensional maps. Although there are different methodologies to generate such maps on computers, the best alternative so far is the geostatistical approach under the umbrella of Kriging method with its basic gradients that describe the spatial dependence over the region of interest.

Natural resources spatial estimation is essential for socio-economic planning, operation, and maintenance of many engineering and earth systems projects. The true value of regional variable is not known at a desired point until either mining or drilling is completed during a field study. The regionalized variable can be considered as a random field, and any data value as the outcome of this field is recorded spatially at a set of sites. In any measurement, there is always a prediction error, which may be significant for the economic planning and decision making. There are different approaches for the regional dependence description either through conceptual geometrical methods or by geostatistical approaches such as Kriging. The advantages and disadvantages of Kriging methods are discussed by various authors (Matheron 1963; Journel and Huijbregts 1978; Sen 2015). Spatial variables are sampled at a set of irregular sites, such as the meteorology stations and groundwater or oil wells. In this chapter rather than the Kriging or any other approach an

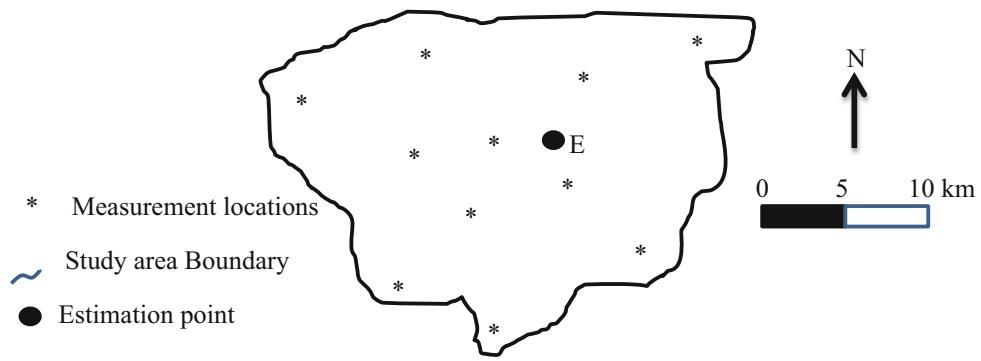
innovative empirical regional dependence function (RDF) is defined from the measurements available at irregularly scattered sites, and this empirical function helps for estimation of the same spatial variable at any desired site within the study area. Each random field has its spatial dependence function (SDF), and the classical spatial continuity is measured by the correlation between the regionalized random variables. The second-order stationary implies that the arithmetic average and standard deviation of the regionalized variable are constants over the whole variability area.

The main purpose of this chapter is to discuss some of the spatial analysis methodological approaches briefly, and then effective MATLAB programs are presented for common use.

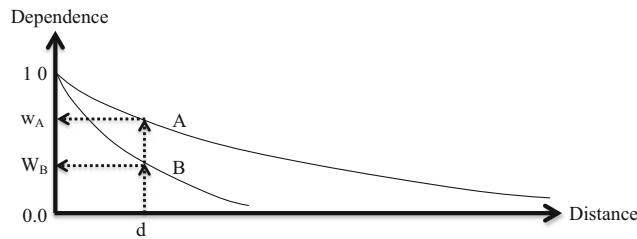
## 8.2 Spatial Dependence Function (SDF)

There are various alternatives to evaluate temporal dependence of a time series (Chaps. 5 and 7) as parametric and nonparametric approaches by correlation measures, but its spatial correspondence is not easy to provide, because the measurement locations are not at equal distances along a line or on a plane or in a space. It is, therefore, necessary to understand and grasp the spatial dependence methodology bases so that one can develop reliable mathematical procedures, equations, and formulations that can be translatable to a computer language, which is the MATLAB software facility in this book.

Prior to formal spatial dependence, it is very useful to understand the fundamentals in the spatial dependence concept. Let us consider that there are a set of measurements at irregularly scattered sites within a study area (Fig. 8.1). In order to simplify the basic concept, let the spatial phenomenon has continuity without any abrupt jump such as the temperature spatial variable.



**Fig. 8.1** Irregular measurement sites



**Fig. 8.2** Spatial dependence functions

If two sites are close to each other, they will have almost the same value, and therefore, the dependence between them is almost complete, which can be represented by a number as 1 (see Fig. 8.2). As the distance increases, it is logical to expect decrease in the dependence between two sites and at large distances the dependence value approaches to zero level.

Logically, the spatial dependence function  $B$  has shorter dependence distance compared to  $A$ . These are the theoretical conceptual models for spatial dependence of a regionalized variable. In the following section, an experimental spatial dependence function (SDF), similar to the one

already explained in the previous section, is proposed as a new weighting function concept. The SDF implies that as the measurement point gets away from the estimation site, its influence in the estimation decreases; i.e., its effective weight gets smaller. For any given distance, say,  $d$  in Fig. 8.2, the weights,  $w_A$  and  $w_B$ , are shown on the vertical axis.

The spatial dependence function calculations can be achieved by the following MATLAB program written by the author. The title of this program implies that SDF can be obtained for each point in terms of other points' distances and measurements. Within this program, MATLAB statement `lat(N)`, `lon(N)`, `lat(i)`, and `lon(i)` computes the distance between points  $N$  and  $i$ .

```

function [RDF,SD]=StationRegionalDependenceFunction(X,lat,lon,N)
% This program calculates Regional Dependence Function, RDF, for a given set of
% monthly data in Saudi Arabia
% X is the monthly data at a set of stations, IMPORTANT NOTE: There are 27 stations
% SD is the sorted distance in ascending order
% N is the desired station number
% First, calculate the distance sequence
k=0;
for i=1:27
    k=k+1;
    d=100*distance(lat(N),lon(N),lat(i),lon(i));
    Dis(k)=d;
end
[SD,I]=sort(Dis);
% Second, calculate the square difference sequence
for im=1:12
    k=0;
    for i=1:27
        k=k+1;
        sq=(X(im,N)-X(im,i))^2;
        SQ(k)=sq;
    end
    % Third, sort SQ in ascending order
    for i=1:k
        j=I(i);
        SSQ(i)=SQ(j);
    end
    % Fourth, calculate cumulative sums
    sum=0;
    for i=1:k
        sum=sum+SSQ(i);
        SSSQ(i)=sum;
    end
    % Fifth, Calculate standard sums, subtract from 1 and save the monthly RDF
    % as MRDF
    for i=1:k
        RDF(im,i)=1-SSSQ(i)/SSSQ(k);
    end
end
end

```

Examples for the SDF are given already in the previous section for climate change downscaling procedure.

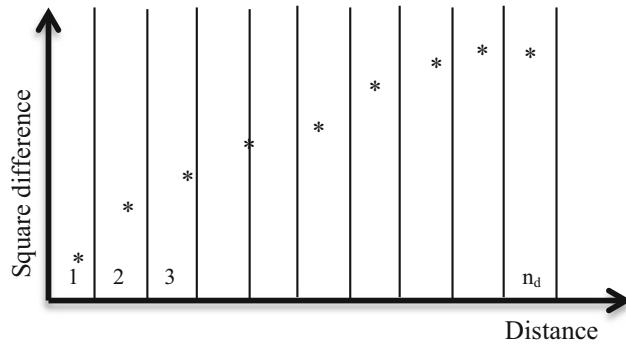
### 8.3 Semi-variogram (SV)

This is a very useful tool for visualization, interpretation, and modeling of spatial variables. A set of sample sites that are irregularly scattered in a study area are selected for measurements with their location coordinates (see Fig. 8.1). Pairwise variability of the phenomenon is sampled by considering the measurement values differences between any two sites. If there are  $n$  sites, then there will be  $n(n - 1)/2$  pairwise differences, and similarly, the same number of distances are available between two stations. For instance, if any two sites, distant  $d$  apart, have measurements  $Z_i$  and  $Z_{i+d}$ , then the measurement difference is  $(Z_i - Z_{i+d})$ , where  $d$  implies distance, and  $i = 1, 2, \dots, n$  number of sites. The summation of all these differences may be equal to zero, which does not provide any information about the regional

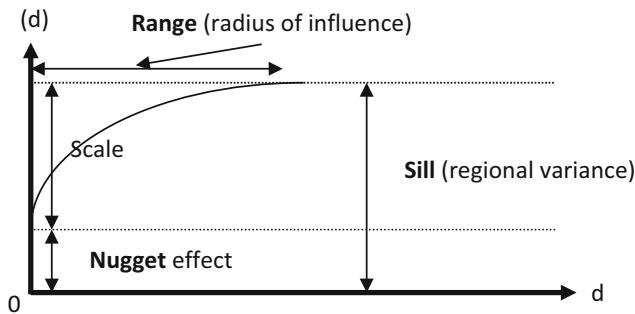
variability. Instead, the difference squares,  $(Z_i - Z_{i+d})^2$ , are adapted for better regional variability representation. Logically, one expects that the more is the distance the bigger is the squared difference in accordance with the reverse of SDF concept (Sect. 8.2). In practical applications, the domain between the minimum and maximum distances is divided into  $n_d$  classes and then squared differences that fall within each graph are averaged leading to the scatter points as in Fig. 8.3.

The classical SV technique is proposed by Matheron (1963) to eliminate the drawbacks in the classical correlation coefficient function, which requires normality (Gaussian distribution) of the observation set and also equal distance requirement. Mathematically, SV,  $\gamma(d)$ , is defined by considering all available sites within the study area as (Clark 1979; Sen 2016),

$$\gamma(d) = \frac{1}{2n_d} \sum_{k=1}^{n_d} (Z_i - Z_{i+d})^2 \quad (8.1)$$



**Fig. 8.3** Experimental SV



**Fig. 8.4** Theoretical SV and elements

where  $k = 1, 2, \dots, n_d = n(n - 1)/2$ . In this expression, instead of  $1/n_d$ ,  $1/2n_d$  is adapted for theoretical convenience, and therefore, Eq. (8.1) is referred to as the semi-variogram. In the calculation of SV, the probability distribution function (PDF) of the spatial variable is not taken into consideration. Different elements of an experimental SV are presented in Fig. 8.4.

Expert reasoning of SV model in this figure helps to elaborate some fundamental points as follows (Şen 2016).

- (1) If the spatial variable is continuous without any discontinuity, then the SV should start from the origin, which means that at zero distance SV is also zero,
- (2) If there is any discontinuity within the spatial variable, then at zero distance a nonzero value appears on the vertical SV axis,
- (3) If there is an extensive spatial dependence, then the SV approaches asymptotically to a constant value at large distances,

- (4) When the spatial dependence is not existent, then the SV has a constant nonzero value equal to the regional variance of the spatial variable at all distances,
- (5) The SV increases as the distance increases until to a certain distance point. Away from this point SV equals the regional variance causing to a flat (stabilization) part on the SV value, which is called as a sill,
- (6) At some distance, called the range, the SV will become approximately equal to the variance (sill) of the spatial variable itself. This is the greatest distance over which the value at a point is related to the value at another point. The range defines the maximum neighborhood over which control points should be selected to estimate a grid node, to take advantage of the statistical correlation among the observations. In the circumstances where the grid node and the observations are spaced so that all distances exceed the range, then Kriging method produces the same estimate as classical statistics, which is equal to the mean value.

The SV calculation for a given set of irregularly located sites can be achieved by the following MATLAB program.

```

function [D,SV]=SemivariogramMATLAB(X, Title)
% THIS PROGRAM FINDS SEMI-VARIOGRAM WITH GROUPING OF DISTANCES
% THE PROGRAM HAS BEEN WRITTEN BY ZEKAI SEN ON 12-TH APRIL, 1986 in FORTRAN
% X.....is a matrix of three columns and N rows. The first two columns are for
%       the longitude and latitude (Easting and Northing) and the third one is
%       for the regional variable records at a set of N sites.
% D.....is the distance vector among all sites
% Title.is the title of the study area
% SV....is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
R=X(:,3); % Regional variable series
MR=mean(R); % Arithmetic average of the regional variable
SR=std(R); % Standard deviation of the regional variable
% Calculate the distances, D, and half square differences, HSD among the sites
k=0;
for i=1:N1
    k=k+1;
    for j=1:N
        D(k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2); % Distances
        HSD(k)=0.5*(X(i,3)-X(j,3))^2; % half square differences
    end
end
[DS,I]=sort(D); % DS sorted distance values and I sorted distance counters
SV=HSD(I);
figure
scatter(DS,SV, '*')
title>Title)
box on
grid on
xlabel('Distance (m)')
ylabel('Semivariogram value')
end

```

The application of this program is given for the data in Table 8.1 for iron ore percentages at 21 sites (Pham 1997).

Figure 8.5 shows the relative positions of each iron ore measurement site, which are scattered irregularly in the study area.

The same data in Table 8.1 is entered into the above MATLAB program for the calculation of the classical semi-variogram diagram, and it gives rise to scatter points in Fig. 8.6.

Theoretically, one expects that the SV scatter points should have a definite trend similar to the one in Fig. 8.4, but the SV points do not indicate such a pattern and are scattered rather randomly. Perhaps the best that can be done is to fit a definite line similar to the one in Fig. 8.4, but no one will agree on such a thing, because the scatter points are far away from any line at many distances. This point indicates that the iron ore percentage original data cannot yield a proper SV. In such cases, the cumulative semi-variogram approach helps to obtain a regular trend.

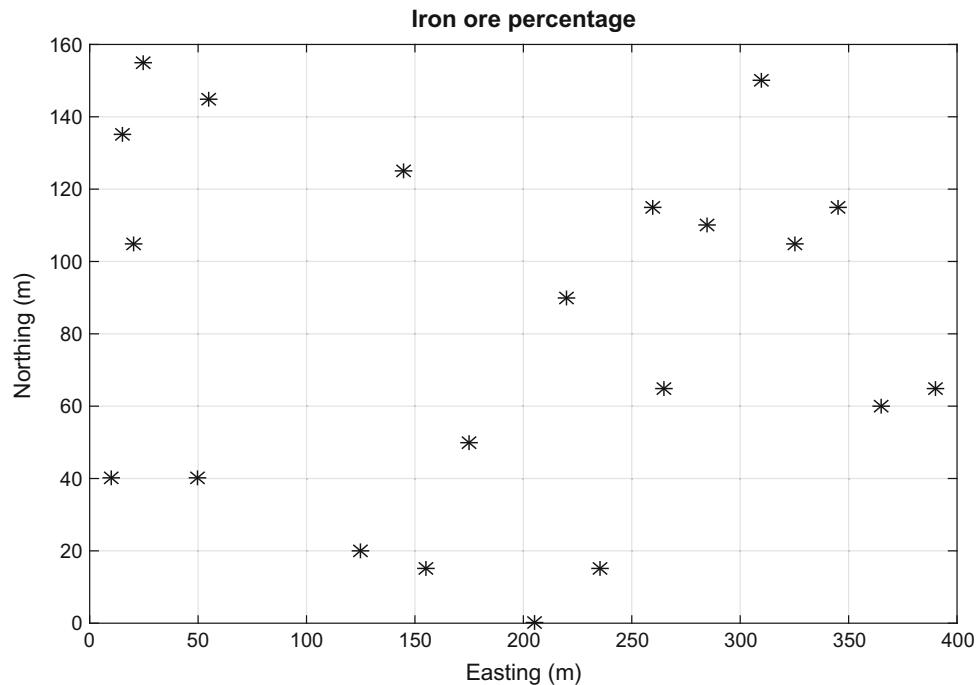
## 8.4 Cumulative SV

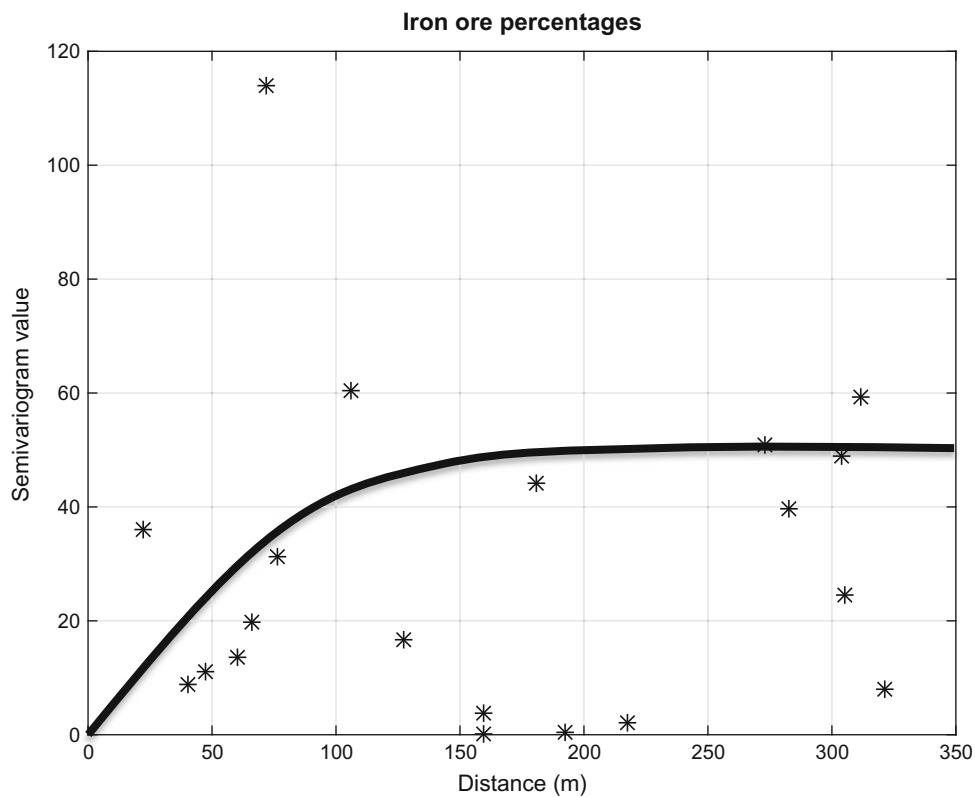
Since general features of the experimental CSVs are already explained by Sen (1989), they are not repeated here. Experimental (Fig. 8.3) and theoretical (Fig. 8.4) SVs are expected to have increasing half square difference values with the distance, but in practical studies SV from the sample measurements may not have such a trend with ups and downs in an irregular manner as in Fig. 8.7.

In the SV graph, whatever are the distances in the plotting distances versus half square differences, the distances appear in their natural sequence in ascending order. In order to get rid of the sample irregularities in the SV, Sen (1989) has suggested the cumulative semi-variogram (CSV) which represents the accumulative half square differences on the vertical axis. In this manner, a non-decreasing pattern is obtained for any set of measurement points, and then a no-decreasing theoretical

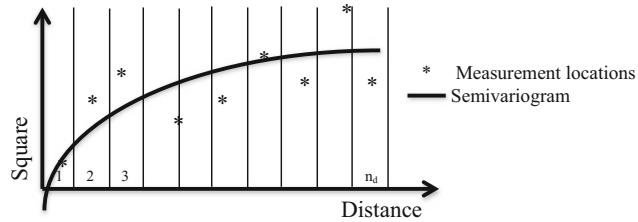
**Table 8.1** Iron ore percentages

Site number	Easting	Northing	%Fe
1	10	40	35.5
2	55	145	29.4
3	175	50	36.8
4	235	15	33.7
5	285	110	35.3
6	20	105	32.5
7	50	40	30.6
8	145	125	30.1
9	205	0	40.1
10	390	65	31.6
11	310	150	34.8
12	15	135	28.6
13	125	20	41.5
14	260	115	33.2
15	365	60	34.3
16	345	115	31.0
17	25	155	29.6
18	155	15	40.4
19	220	90	28.5
20	265	65	24.4
21	325	105	39.5

**Fig. 8.5** Iron ore sampling positions



**Fig. 8.6** Classical semivariogram scatter points



**Fig. 8.7** SV irregularities

function is matched to the CSV. Finally, the differential of this theoretical curve with respect to distance yields SV theoretical graph.

The following MATLAB program provides an electronic computation procedure for the CSV from a set of irregularly scattered measurement sites.

```

function [DS,CumSV] = CumulativeSemivariogram(X,Title)
% This program calculates cumulative semivariogram
% X is the data with three columns: The first two columns include the
% easting and northing of the site locations and the third column is the
% regional measurements
% Title is the title of the study area of the CumSV
% DS is the sorted distance
% CumSV is the output of the program as Cumulative semivariogram
N=length(X(:,1));
% Calculate the distances among the sites
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            D(k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2);
        else
        end
    end
end
% Calculate the square differences
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            SQ(k)=((X(i,3)-X(j,3))^2);
        else
        end
    end
end
[DS,I]=sort(D);
SQS=SQ(I);
CumSV=cumsum(SQS);
plot(DS,CumSV,'*')
xlabel('Distance (m)')
ylabel('Cumulative semivariogram')
title>Title)
box on
grid on
end

```

The application of this MATLAB program to the available data in Table 8.1 leads to Fig. 8.8.

It is obvious that the CSV provides a definite trend and it can be regarded as the integration of the classical SV. Consequently, after the matching a theoretical mathematical function to the scatter points, one can take the derivative to reach at a proper SV expression.

## 8.5 Point CSV

This is similar to CSV, but it does not take into consideration all the possible distances among the measurement sites. It is based on the distances between a fixed site and others for the point CSV (PCSV) calculations. Hence, instead of  $n$   $(n - 1)/2$  distances there are only  $(n - 1)$  distances. Each measurement point has its own PCSV pattern and corresponding point regional dependence function (PRDF) (Şen and Habib 1998). The PCSV identifies the spatial variable

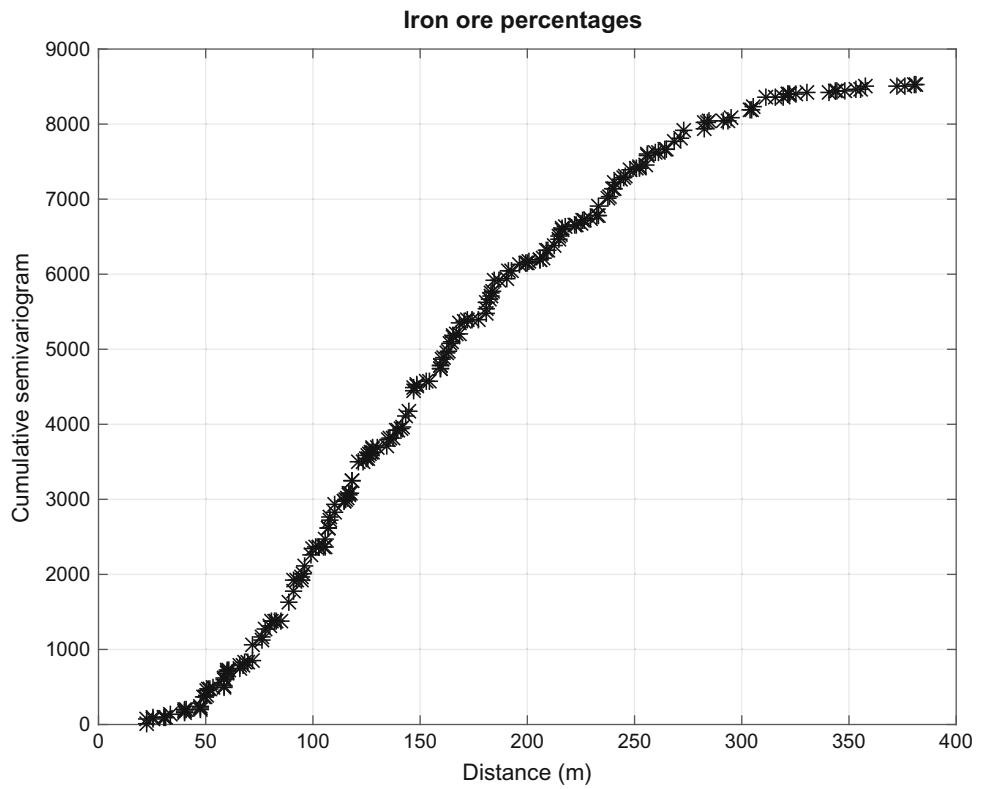
variability around a single site rather than the whole region. The treatment of available spatial data at  $n$  sites according to the following steps leads to sample PCSV for a particular site (Şen 2016).

- (1) Let  $\bar{Z}$  and  $Z_s$  be the arithmetic average and standard deviation, of the regional measurement data, respectively. The standardized regional data measurement values,  $z_i$ , are calculated from the actual regional measurements,  $Z_i$  as,

$$z_i = \frac{Z_i - \bar{Z}}{Z_s} \quad (8.3)$$

- (2) The  $n - 1$  distances are calculated,  $d_i$  ( $i = 1, 2, \dots, n - 1$ ) between a site and others,
- (3) The square standard measurement differences as  $(z_E - z_i)^2$  are calculated, where  $z_E$  and  $z_i$  are the regional variable values at the estimation and  $i$ -th sites, respectively,

**Fig. 8.8** CSV for iron ore percentages



- (4) Plot of  $d_i$  versus corresponding successive cumulative sums of half-squared differences and, hence, the sample PCSV,  $\gamma(d_c)$ , can be obtained for the concerned site as follows,

$$\gamma(d_c) = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} (z_C - z_i)^2 \quad (8.4)$$

In this manner,  $n - 1$  PCSV empirical patterns can be obtained each for a measurement site. No need to say that

one can obtain also the PRDF similar to the SDF identification from the PCSV concept as explained in Sect. 8.2.

Comparison among the PCSVs provides rich information about the behavior of the spatial variable concerned. It is possible to determine the radius of dependence influence for each site. Hence, one can appreciate the structural behavior of the regional or spatial variable near the site such as the nugget (sudden changes) and sill effects and heterogeneity.

The following MATLAB program is for the PCSV calculation for each measurement site.

```

function PointCumulativeSemivariogram(X,Title)
% This program is written by Zekai Sen, Istanbul Technical University
% on 29 July 2013
% X.....is a matrix of three columns and N rows. The first two columns are for
%      the longitude and latitude (Easting and Northing) and the third one is
%      for the regional variable records at a set of N sites.
% DS....is the distance vector among all sites
% Title.is the title of the study area
% SV....is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
R=X(:,3); % Regional variable series
MR=mean(R); % Arithmetic average of the regional variable
SR=std(R); % Standard deviation of the regional variable
% Calculate the distances, D, and half square differences, HSD between the
% pivot site and others
for i=1:N1
    k=0;
    for j=1:N
        k=k+1;
        D(i,k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2); % Distances from pivot site
        HSD(i,k)=0.5*(X(i,3)-X(j,3))^2; % Half square differences from pivot site
    end
end
% Calculate the PCSV for each site%
for i=1:N1
    [DS I]=sort(D(i,:)); % Sorted distances in ascending order
    SPCSV=HSD(i,I); % Accordingly arranged half square differences
    CumPCSV=cumsum(SPCSV);
    figure
    scatter(DS,CumPCSV, 'k*')
    box on
    grid on
    title>Title
    xlabel'Distance (m)'
    ylabel'Point cumulative semivariogram'
    title>Title
end
end

```

The application of this MATLAB program to data in Table 8.1 leads to 21 PCSV scatter diagram. Only six of them are given in Fig. 8.9.

One can see that each one of the PCSV scatter diagrams in this figure has non-decreasing trend and they can be matched with a suitable mathematical formulation to obtain the general trend (Sen 2015). Each one of these PCSVs is useful in obtaining the PRDF for the site concerned with the identification of radius of influence as explained in Sect. 8.1.

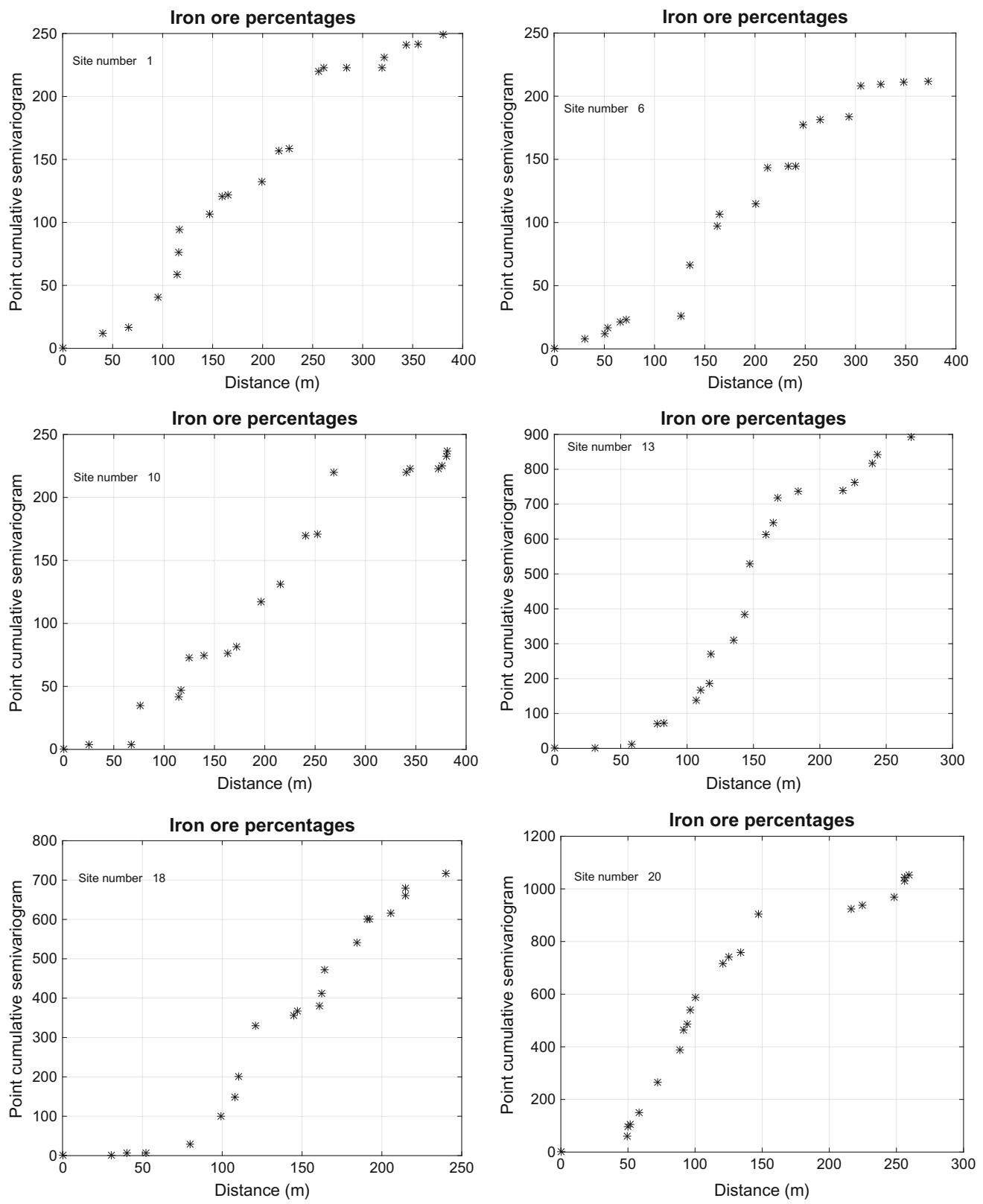
## 8.6 Directional Semi-variogram (DSV)

All what have been explained in all the previous sections, and the configurations and their consequent regional reflections in the calculations did not take into consideration the variations along a specific direction; hence, they were all isotropic measure or omni-directional globally. In some cases, the direction of the phenomenon can be important like the stream flow, highway, and line source of pollutions. The SVs based on a set of directions (directional semi-variogram, DSV) provide information as to the isotropy or anisotropic

of the regional variable. If SVs along different directions are not significantly different from each other say, within  $\pm 5\%$  or  $\pm 10\%$  relative error bands, then the concerned regional variable has isotropic structure, otherwise it is anisotropic.

The regularity and continuity of the regional variable of a natural phenomenon are represented by the behavior of SV near the origin as sill value (Fig. 8.2). Furthermore, SV plots along different directions provide valuable information about continuity and homogeneity. A properly fitted theoretical SV model allows linear estimation calculations that reflect the spatial extent and orientation of spatial dependence in the regional variable map (Davis 1986; Clark 1979). Without a rationale for identifying the major direction of anisotropy, the following steps might be useful in narrowing the focus of the exercise (Sen 2016).

- (1) Begin by considering an omni-directional SV with a bandwidth large enough to encompass all data points on the site. In practice, maximum lag distance can be taken as 1/3 of the maximum distance between the data points,
- (2) Select the number of lags and lag distances sufficient to span a significant portion of the entire site, and choose

**Fig. 8.9** PCSV scatter diagrams

- the lag tolerance to be very close in value to the lag distance itself,
- (3) Calculate the SV. In most cases, data become less correlated as the distance increases between them. Under these circumstances, the SV values should produce a monotonic increasing function, which approaches a maximal value called the sill. In practice, this may not be the case with SV values that may begin high value or jump around as distance increases,
  - (4) Adjust the number of lags and lag tolerances until, generally, a monotonically increasing trend is seen in the SV values. If this cannot be achieved, it may be that a geostatistical approach is not viable or that more complicated trends occur than can be modeled. If a visual inspection of the data or knowledge about the dispersion of contamination indicates a direction of correlation, it may be more appropriate to test first this direction,

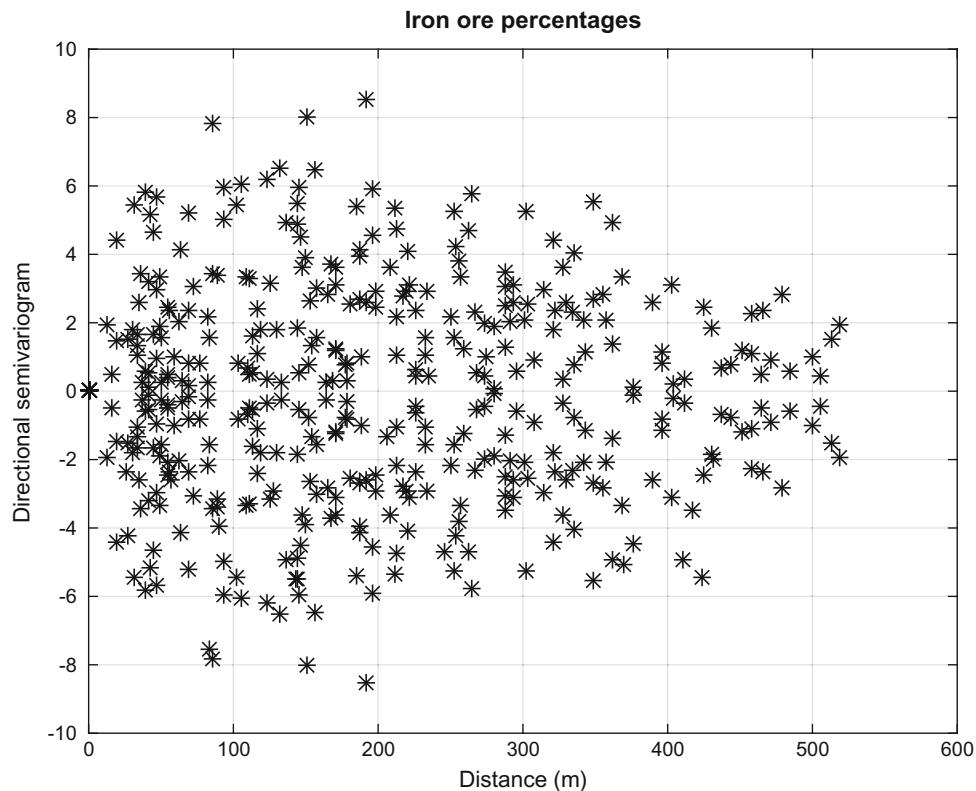
- (5) Assuming the omni-directional SV adds reasonably another direction to the plot with a smaller tolerance, one may have to adjust the bandwidth and angle tolerance to produce a reasonable SV plot,
- (6) If the second direction rises slower to the sill or rises to a lower sill, then this is the major direction of anisotropy,
- (7) If neither direction produces significantly lower spatial correlation, it may be reasonable to assume an isotropic correlation structure,
- (8) Add a cone structure with direction equal to the major direction plus  $90^\circ$ , and model the SV results in this direction,
- (9) If the data are isotropic, choose the omni-directional SV as the major direction.

The following MATLAB program helps to calculate semi-variogram along different directions.

```

function [SD,DSV]=DirectionalSemivariogram(X,theta,Title)
% X.....is a matrix of three columns and N rows. The first two columns are for
%       the longitude and latitude (Easting and Northing) and the third one is
%       for the regional variable records at a set of N sites.
% Theta. Desired direction angle in radian
% DS....is the distance vector among all sites
% Title.is the title of the study area. It must be entered in radians such
%       as pi, pi/2, pi/6, or any other direction
% DSV...is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
for i=1:N
    Dx(i)=X(i,1)*cos(theta)+X(i,1)*sin(theta); % Easting values projection on desired
direction
    Dy(i)=-X(i,2)*sin(theta)+X(i,2)*cos(theta); % Northing values projection on the desired
direction
end
% Distance and half square difference calculations
k=0;
for i=1:N1
    for j=1:N
        k=k+1;
        D(k)=sqrt((Dx(i)-Dx(j))^2+(Dy(i)-Dy(j))^2); % Distances
        HSD(k)=0.5*(X(i,3)-X(j,3)); % Half square differences
    end
end
% Sort distances in ascending order
[SD I]=sort(D);
% Arrange the half square differences according to sorted distances
DSV=HSD(I);
% Directional semi variogram plot
figure
plot(SD,DSV,'k*')
title>Title)
xlabel('Distance (m)')
ylabel ('Directional semivariogram')
box on
grid on
end

```



**Fig. 8.10** DSV along  $\pi/3$  direction

Again with the same data in Table 8.1, the directional semi-variogram along  $\pi/3$  direction is given in Fig. 8.10.

Similar to SV in Fig. 8.6, there is not a definite pattern but haphazard scatter diagram for DSV. It is possible to

convert previous MATLAB program to cumulative and also point SV versions. In the following cumulative DSV program is presented.

```
function [SD,DSV]=CumulativeDirectionalSemivariogram(X,theta,Title)
% X.....is a matrix of three columns and N rows. The first two columns are for
%   the longitude and latitude (Easting and Northing) and the third one is
%   for the regional variable records at a set of N sites.
% Theta. Desired direction angle in radian
% DS....is the distance vector among all sites
% Title.is the title of the study area. It must be entered in radians such
%   as pi, pi/2, pi/6, or any other direction
% DSV...is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
for i=1:N
    Dx(i)=X(i,1)*cos(theta)+X(i,2)*sin(theta); % Easting values projection on the desired
    direction
    Dy(i)=-X(i,2)*sin(theta)+X(i,2)*cos(theta); % Northing values projection on the desired
    direction
end
% Distance and half square difference calculations
k=0;
for i=1:N1
```

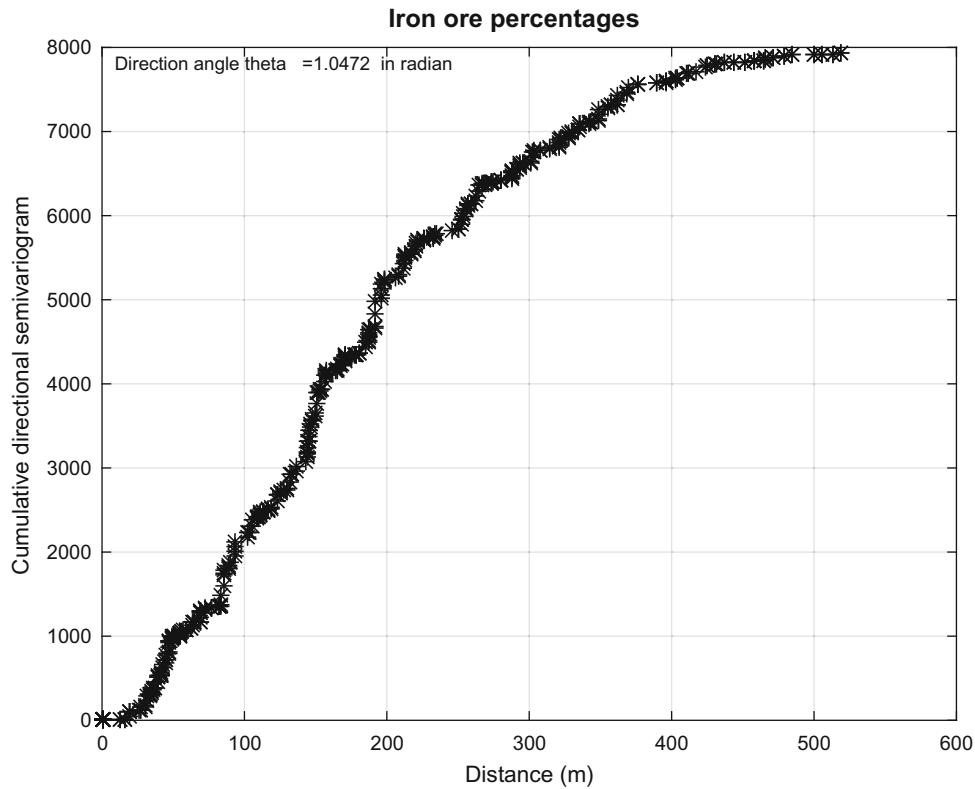
```

for j=1:N
    k=k+1;
    D(k)=sqrt((Dx(i)-Dx(j))^2+(Dy(i)-Dy(j))^2); % Distances
    HSD(k)=0.5*(X(i,3)-X(j,3))^2; % Half square differences
end
% Sort distances in ascending order
[SD I]=sort(D);
% Arrange the half square differences according to sorted distances
DSV=HSD(I);
% Find cumulative half squared differences
CumDSV=cumsum(DSV);
% Directional semi variogram plot
figure
plot(SD,CumDSV,'k*')
title('Title')
xlabel('Distance (m)')
ylabel ('Cumulative directional semivariogram')
box on
grid on
x=min(SD)+10;
y=max(CumDSV)-100;
text(x,y,['Direction angle theta =',num2str(theta),' in radian'])
end

```

The same data in Table 8.1 and arrangement as in the previous MATLAB program cumulative directional semi-variogram appears as in Fig. 8.11 in a definite pattern along  $\pi/3$  direction.

Finally, one can obtain also point cumulative directional SV through a similar MATLAB program to the last two MATLAB writings as follows.



**Fig. 8.11** Cumulative DSV

```

function [SD,PCumDSV]=PointCumulativeDirectionalSemivariogram(X,theta,Title)
% X.....is a matrix of three columns and N rows. The first two columns are for
%      the longitude and latitude (Easting and Northing) and the third one is
%      for the regional variable records at a set of N sites.
% Theta. Desired direction angle in radian
% DS....is the distance vector among all sites
% Title.is the title of the study area. It must be entered in radians such
%      as pi, pi/2, pi/6, or any other direction
% DSV...is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
for i=1:N
    Dx(i)=X(i,1)*cos(theta)+X(i,1)*sin(theta); % Easting values projection on the desired direction
    Dy(i)=-X(i,2)*sin(theta)+X(i,2)*cos(theta); % Northing values projection on the desired direction
end
% Distance and half square difference calculations
for i=1:N1
    k=0;
    for j=1:N
        k=k+1;
        D(i,k)=sqrt((Dx(i)-Dx(j))^2+(Dy(i)-Dy(j))^2); % Distances
        HSD(i,k)=0.5*(X(i,3)-X(j,3))^2; % Half square differences
    end
end
% Plot the scatter of point cumulative SVs
for i=1:N1
    % Sort distances in ascending order
    [SD,I]=sort(D(i,:));
    % Arrange the half square differences according to sorted distances
    DSV=HSD(i,I);
    % Find cumulative half squared differences
    PCumDSV=cumsum(DSV);
    % Directional semi variogram plot
    figure
    plot(SD,PCumDSV,'k*')
    title>Title
    xlabel'Distance (m)'
    ylabel 'Point cumulative directional semivariogram'
    box on
    grid on
    x=min(SD)+10;
    y=max(PCumDSV)-10;
    text(x,y,['Direction angle theta =',num2str(theta), ' in radian', 'Site number =',num2str(i)])
end
end

```

Figure 8.12 provides six directional point cumulative semi-variogram scatter graphs from data given in Table 8.1.

Compared with the DSV graphs, the point cumulative directional SVs provide definitive patterns similar to the CSV in Fig. 8.8.

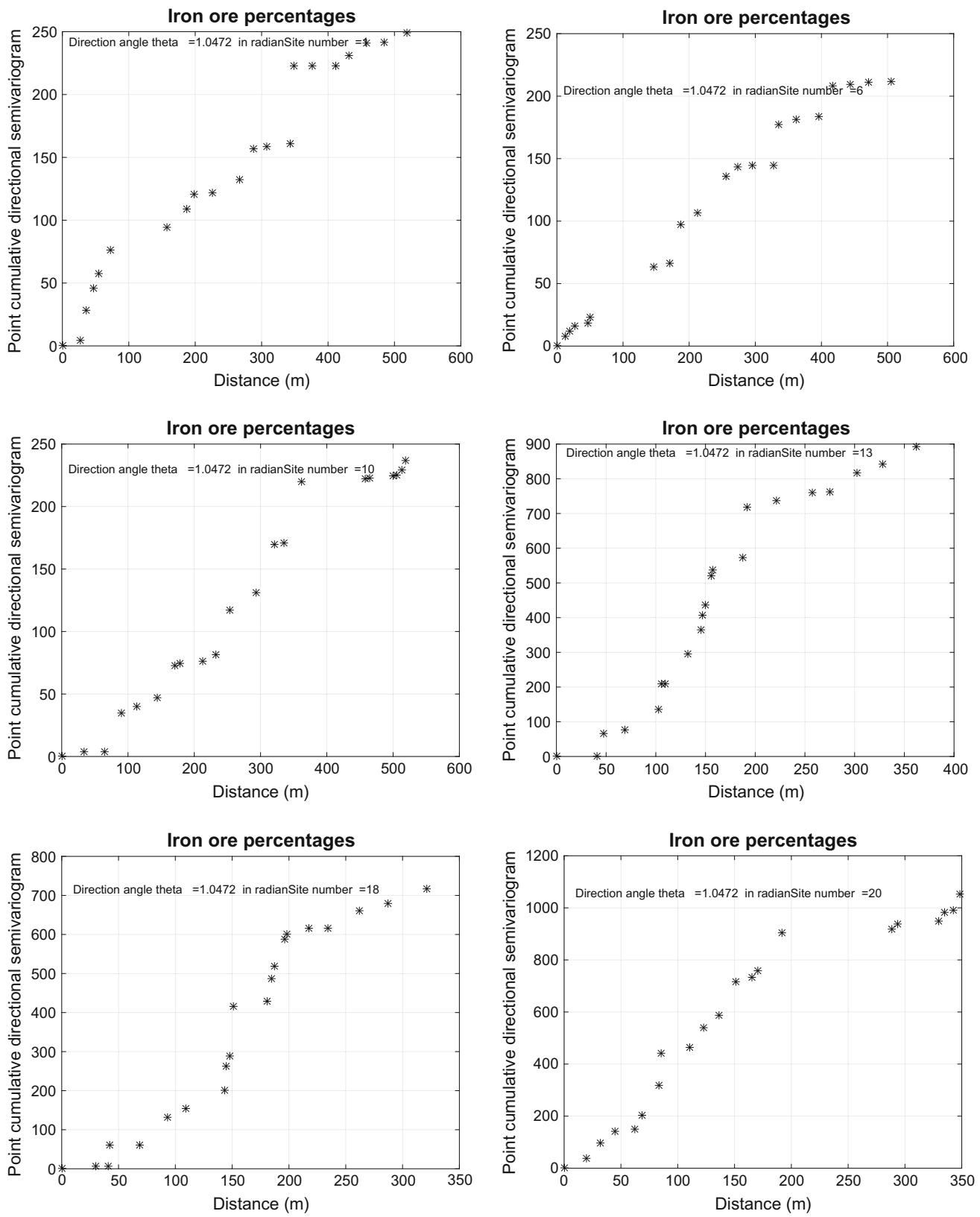
## 8.7 Innovative Spatial Dependence Function

The theoretical background for the SDF is given in Sect. 8.2. Its construction from a set of regionalized variable measurements can be obtained in an innovative manner from the CSV through the execution of the following steps.

- (1) The CSV empirical value,  $\gamma_{c_d^*}$  at a large distance,  $d^*$  is decided such that its difference from the next one is within a certain relative error limit of  $\pm 10\%$  or better  $\pm 5\%$ . This

implies that these two points are practically at the end of empirical CSV, and any distance more than this limit means that there is no more regional dependence. The  $\gamma_{c_d^*}$  is regarded as the maximum CSV value,

- (2) Whole CSV ordinates  $\gamma_{c_d}$  are divided by the maximum value, which is referred to as the ratio,  $r = \gamma_{c_d} / \gamma_{c_d^*}$ , which varies between 0 and 1. Hence, the final product is standard CSV as non-decreasing pattern for the regionalized variable,
- (3) The SDF is obtained by subtraction of the ratio values from 1 at each distance. This time a steadily decreasing pattern of weight  $w_d = 1 - r = 1 - \gamma_{c_d} / \gamma_{c_d^*}$  is obtained,
- (4) It is now possible to attach any distance with the weight value. It is obvious that the larger is the distance between two points the smaller is the weight value, hence the regional correlation coefficient. The weights can be regarded as the regional correlation coefficients.



**Fig. 8.12** Point cumulative DSV graphs

The SDF pattern can be obtained from a set of irregularly scattered site measurements by means of the following MATLAB program.

One can also obtain site-specific SDF, which is very useful in making the estimation of the site measurement for verification of the regional estimation procedure.

```

function [DS,RDF] = RegionalDependenceFunction(X)
% This program calculates cumulative semivariogram
% X is the data with three columns: The first two columns include the
% easting and northing of the site locations and the third column is the
% regional measurements
% Title is the title of the study area of the CumSV
% DS is the sorted distance
% RDF is the values of spatial dependence
N=length(X(:,1));
% Calculate the distances among the sites
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            D(k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2);
        else
        end
    end
end
% Calculate the square differences
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            SQ(k)=((X(i,3)-X(j,3))^2);
        else
        end
    end
end
[DS,I]=sort(D);
SQS=SQ(I);
CumSV=cumsum(SQS);
% Calculate dimensionless CSV
DM=max(DS); % Maximum distance value
CSVM=max(CumSV); % Maximum CSV value
DD=DS/DM;
DCSV=CumSV/CSVM;
% Subtraction of the dimensionless CSV values from one
M=length(DS); % Number of distances
for i=1:M
    RDF(i)=1-DCSV(i); % RDF values
end
figure
scatter(DS,RDF,'k*')
xlabel('Distance (m)')
ylabel('Regional dependence weights')
title('Regional dependence function')
box on
grid on
end

```

The calculation based on data in Table 8.1 leads with the use of the above MATLAB program to SDF as in Fig. 8.13.

The following MATLAB program calculates SDF for each site.

```

function SiteRegionalDependenceFunction(X)
% This program is written by Zekai Sen, Istanbul Technical University
% on 29 July 2013
% X.....is a matrix of three columns and N rows. The first two columns are for
%       the longitude and latitude (Easting and Northing) and the third one is
%       for the regional variable records at a set of N sites.
% DS....is the distance vector among all sites
% Title.is the title of the study area
% SV....is the corresponding semivariogram value(half square differences)
N=length(X(:,1));
N1=N-1;
R=X(:,3); % Regional variable series
MR=mean(R); % Arithmetic average of the regional variable
SR=std(R); % Standard deviation of the regional variable
% Calculate the distances ,D, and half square differences, HSD between the
% pivot site and others
for i=1:N1
    k=0;
    for j=1:N
        k=k+1;
        D(i,k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2); % Distances from pivot site
        HSD(i,k)=0.5*(X(i,3)-X(j,3))^2; % Half square differences from pivot site
    end
end
% Calculate the PCSV for each site%
for i=1:N1
    [DS I]=sort(D(i,:)); % Sorted distances in ascending order
    SPCSV=HSD(i,I); % Accordingly arranged half square differences
    CumPCSV=cumsum(SPCSV);
    MD=max(DS); % Maximum distance value
    MSPCSV=max(CumPCSV); % Maximum of the point CSV
    % Calculate standard PCSV
    DD=DS/MD;
    DSPCSV=CumPCSV/MSPCSV;
    % Subtract standard PCSV values from 1
    M=length(DS); % Number of distances
    for i=1:M
        PRDF(i)=1-DSPCSV(i);
    end
    figure
    scatter(DS,PRDF,'k*')
    box on
    grid on
    xlabel('Distance (m)')
    ylabel('Point RDF value')
    title('Point regional dependence function')
    x=min(DD)+0.1;
    y=max(DSPCSV)-0.1;
    text(x,y,['Site number ',num2str(i)])
end
end

```

The application of the site-specific SDF MATLAB program above to data in Table 8.1 leads to a set of site SDF out of which six are presented in Fig. 8.14.

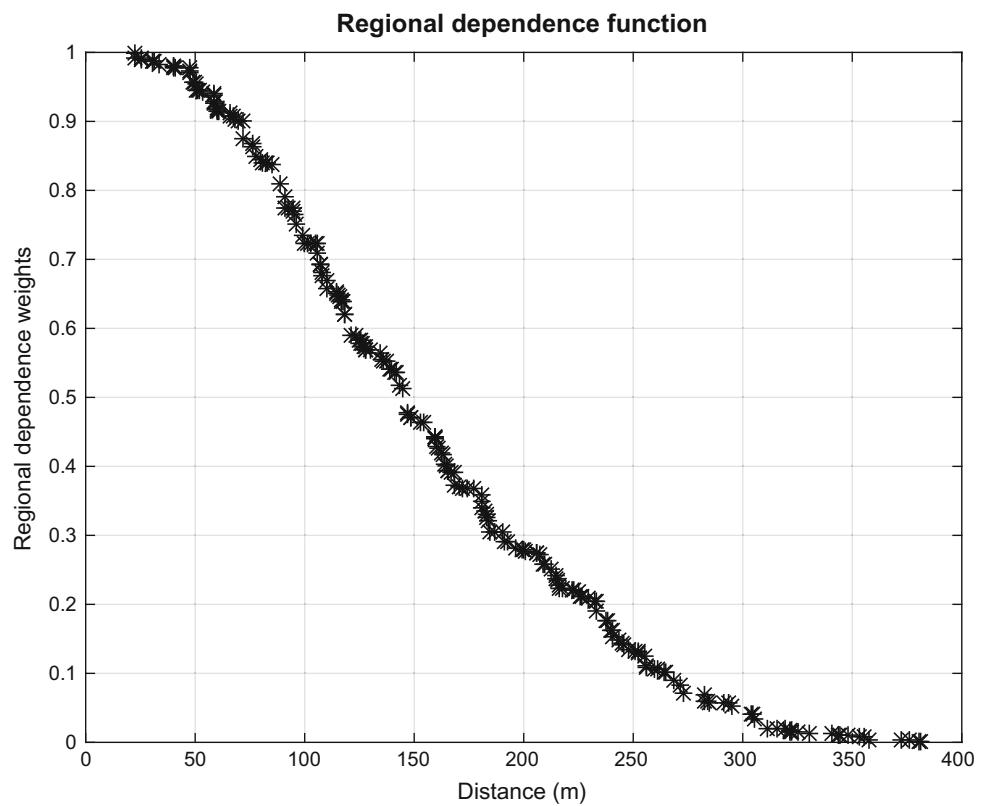
## 8.8 Regional Estimation Procedure

After the identification of SDF, it can be used for regional estimation in a simple manner by using the weights. In Fig. 8.1, there are measurement and estimation points, where there is no record at point  $E$ . It is necessary to estimate the value at this point by means of the measurement values at

the surrounding points. After all what have been explained above, it is well known that the closer is the measurement point to the estimation point the more contributive its value to overall point estimation,  $E$ .

Let the number of the measurement sites  $n$  with an estimation site,  $E$ . It has already been mentioned that the weights are inversely related to distance between each pair of sites. Let the measurements are indicated by  $M_i$  ( $i = 1, 2, \dots, n$ ) and the estimation value at point  $E$  with  $E_V$ . Under the light of explanations for the SDF estimation of regionalized variable at  $E$  can be achieved as a weighted average of the measurements as,

**Fig. 8.13** Global regional dependence functions



$$E_V = \frac{\sum_{i=1}^N w(r_{i,E}) M_i}{\sum_{i=1}^N w(r_{i,E})} \quad (8.2)$$

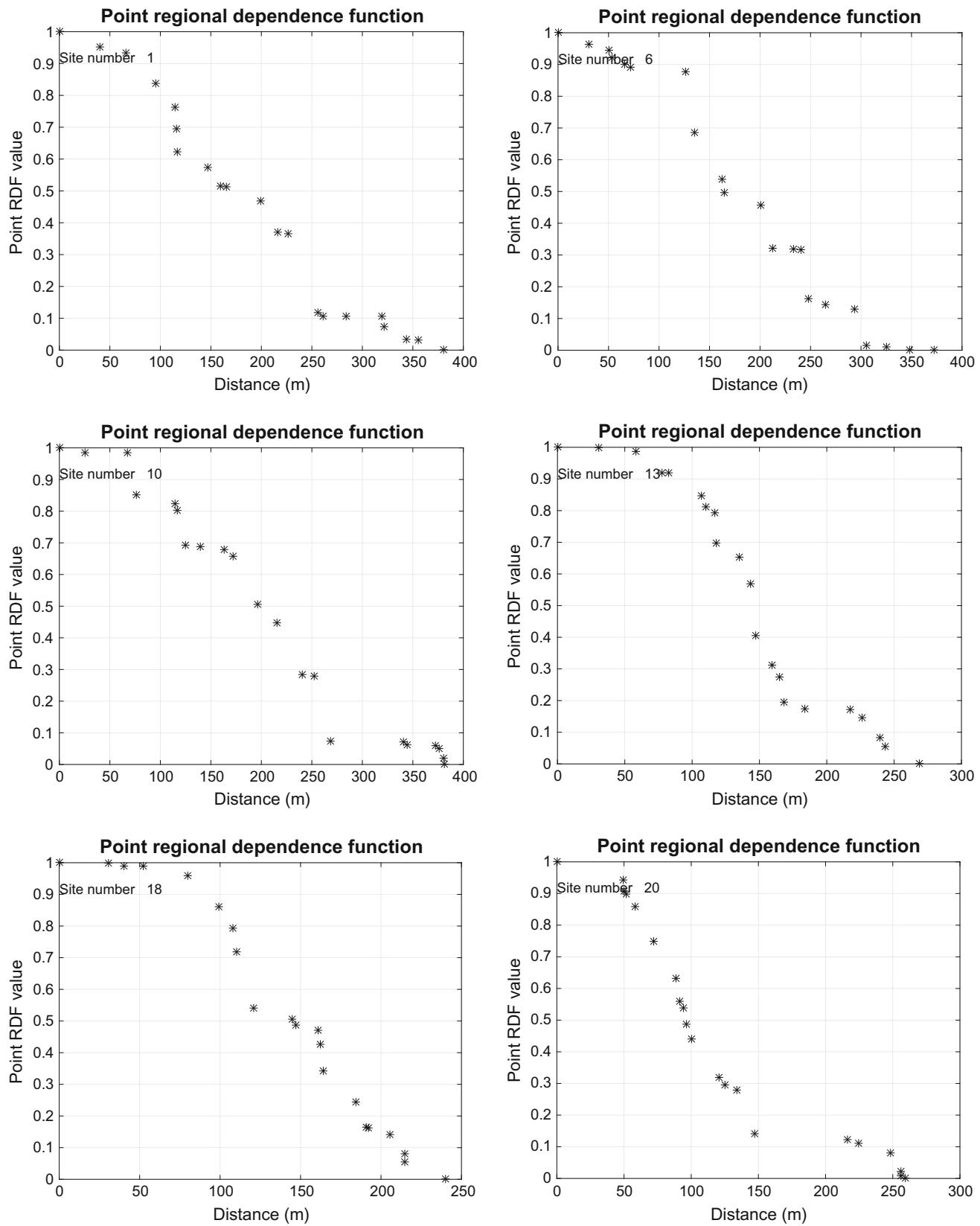
where  $r_{i,M}$  represents the distance between any measurements station  $i$  and estimation site,  $E$ ;  $w(r_{i,E})$  are the weights corresponding to the distance.

According to Clark (1979), the SDF is equivalent to the well-known “inverse square distance (IDS)” procedure, where the distances among the measurement sites are taken into consideration, but actual regionalized variable measurements are overlooked. Similar to the IDS procedure, many of the geometrically suggested logical and conceptual SDFs are not dependent on actual measurements (Cressman

1959; Sasaki 1960; Barnes 1964). In general, the major drawbacks of the weighting functions available in the literature are given by Sen (1997).

Kriging estimator finds optimum set of weights provided that a well-defined semi-variogram is available. In practical applications, usually one cannot find sufficient data to identify a representative semi-variogram. Bardossy et al. (1990a, b) indicated that when sampling sites are inadequate for establishing a semi-variogram, kriging estimates are not reliable.

The radius of influence,  $R$ , of any SDF is the distance after which the regional dependence, i.e., regional correlation coefficient, is equal to zero. The MATLAB programs in the previous sections yield SV, CSV, PCSV, and RDF.



**Fig. 8.14** Site-specific SDFs

```

function [E,W] =SpatialEstimation(X)
% This program calculates cumulative semivariogram
% X: The data with three columns: The first two columns include the
% easting and northing of the site locations and the third coilmn is
% the regional measurements
% DS: The sorted distance that should come from
% "RegionalDependenceFunction" program
% RDF: The values of spatial dependence that should come from
% "RegionalDependenceFunction" program
% W Weights from the regional dependence function that are dependent
% on distances
% E: The spatial estimation values
% Title:The title of the study area
N=length(X(:,1));
% Calculate the distances among the sites
% Calculate the distances among the sites
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            D(k)=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2);
        else
        end
    end
end
% Calculate the square differences
k=0;
for i=1:N
    for j=i:N
        if i ~= j
            k=k+1;
            SQ(k)=(X(i,3)-X(j,3))^2;
        else
        end
    end
end
[DS,I]=sort(D);
SQS=SQ(I);
CumSV=cumsum(SQS);
% Calculate dimensionless CSV
DM=max(DS); % Maximum distance value
CSVM=max(CumSV); % Maximum CSV value
DD=DS/DM;
DCSV=CumSV/CSVM;
% Subtraction of the dimensionless CSV values from one
M=length(DS); % Number of distances
for i=1:M
    RDF(i)=1-DCSV(i); % RDF values
end
figure
scatter(DS,RDF,'k*')
xlabel('Distance (m)')
ylabel('Regional dependence weights')
title('Regional estimation')
box on
grid on
% Calculate spatial estimation weights
for i=2:N
    D=sqrt((X(1,1)-X(1,2))^2+(X(i,1)-X(i,2))^2);
    [DD I]=min(abs(DS-D));
    W1=RDF(I);
    W2=RDF(I-1);
    W(i)=(W1+W2)/2;
end
% Calculate regional estimation
for i=1:N-1

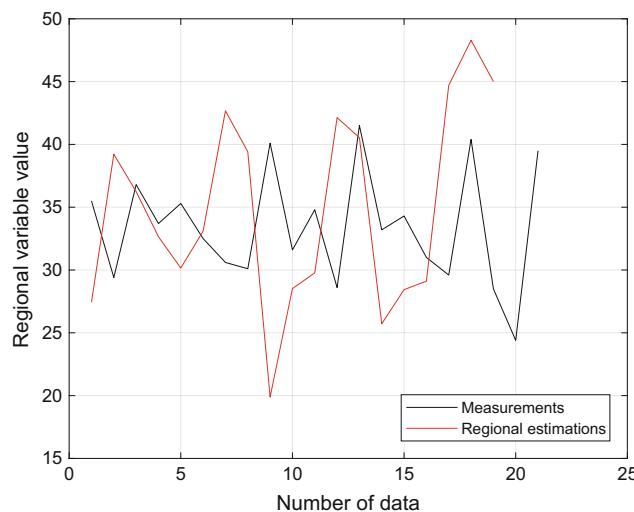
```

.....Continued.....

```

i1=i+1;
for j=2:N
    if i ~= j
        D=sqrt((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2);
        DD=DS-D;
        [M J]=min(abs(DD));
        if J ~= 1
            W1=RDF(J-1);
            W2=RDF(J);
        else
            W=(W1+W2)/2;
        end
    end
end
% Calculate successive estimation for each site
for i=1:N-1
    SN=0;
    SD=0;
    for j=2:N
        SN=SN+X(j,3).*W(i,j);
        SD=SD+W(i,j);
        E(i,j)=SN/SD;
    end
end
figure
plot(X(:,3),'k')
hold on
AE=mean(E');
plot(AE(2:end),'r')
box on
grid on
xlabel('Number of data')
ylabel('Regional variable value')
legend('Measurements','Regional estimations','Location','SouthEast')
end

```



**Fig. 8.15** Regional estimation

The application of the above MATLAB program to data given in Table 8.1 yields the regional average estimations at each station as in Fig. 8.15.

## References

- Bardossy, A., Bogardi, I., and Kelly, W.E., (1990a). Kriging with imprecise variograms I: Theory. *Math. Geol.*, Vol. 22(1): 63–79.
- Bardossy, A., Bogardi, I., and Kelly, W.E., (1990b). Kriging with imprecise variograms II: Application. *Math. Geol.*, Vol. 22(1): 81–93.
- Barnes S.L., (1964). A technique for maximizing details in numerical weather map analysis. *J. Appl. Meteor.*, Vol. 3: 369.
- Clark, I., (1979). Practical Geostatistics. Applied Science Publishers Ltd., London, pp: 129.
- Cressman, G.D., (1959). An operational objective analysis concept and application in Turkey. *Solar Energy*, Vol. 68(1): 367.
- Davis, J.C., (1986). Statistics and Data Analysis in Geology, 2nd edition, John Wiley, Chichester, 646 pp.
- Journel, A.G., and Huijbregts, C.J., (1978). Mining Geostatistics. London, San Diego: Academic Press, Harcourt Brace & Company.
- Matheron, G., (1963). Principles of Geostatistics. *Economic Geology* 58, 1246–1266.
- Pham, T.D., (1997). Grade estimation using fuzzy-set algorithms. *Mathematical Geology*, Vol. 29: 291–305.
- Sasaki, Y., (1960). An objective analysis for determining initial conditions for the primitive equations. In Tech. Report, Vol. Ref. 60–16T, Texas A/M University, College Station.
- Şen, Z., (1989). Cumulative semivariogram model of regionalized variables. *Int. J. Math. Geol.*, Vol. 21: 891.
- Şen, Z., (1997). Objective analysis of cumulative semivariogram technique and its application in Turkey. *J. Appl. Meteorol.* Vol. 36:1712–1720.
- Şen, Z., (2015). Spatial Modeling Principles in Earth Sciences. Springer, Second Edition, 403 pp.
- Şen, Z., (2016) Spatial Modeling Principles in Earth Sciences. Springer, Second Edition: 351 pp.
- Şen, Z., and Habib, Z.Z., (1998). Point cumulative semivariogram of areal precipitation in mountainous regions. *Journal of Hydrology*, Vol. 205(1–2): 81–91.

# Renewable Energy

# 9

## 9.1 General

Renewable energy sources are available in any part of the globe in different or combined forms that need attention for refined potentiality calculation methodologies. In the nineteenth century apart from the coal the most exploitative form appeared as hydropower electrical energy, which depended on the water head elevation behind high dams, where the potential energy of water storage is converted to the kinetic energy readily by means of the conversion of this kinetic energy to electrical power. Recently, other types of renewable energy started to take place in the market with productive implementations among which are the wind and solar energy sources. Although there are other types such as wave, tidal, biomass, they are not explained in this chapter. The importance of the renewable energy sources has gained speedy increase after the recognition of global warming, climate change impacts on the human life existence atmospheric environment due to greenhouse gas emissions especially with extensive burning and combustion of fossil fuels including coal, lignite, and oil. The fossil fuels emit especially carbon dioxide into the atmosphere. Its chemical composition was at the level of 288 ppm before the industrial revolution, but today it is more than 400 pp, which implies a significant chemical change in the atmospheric composition. Renewable energy sources are in great demand to replace the fossil fuels mainly for reduction of the greenhouse emissions into the atmosphere.

Various MATLAB language written programs are presented for solar, wind, and current clean energy calculations.

## 9.2 Solar Energy

Solar radiation is an integral part of different renewable energy resources, and it is the main and daylight continuous input variable from inexhaustible sun. Estimation of terrestrial global irradiation amounts from simply measured sunshine durations through pyranometers is a prerequisite in

many social, economic, industrial, and engineering activities. Design of many technical apparatus such as coolers, heaters, solar energy generators in forms of photovoltaics all requires basic data for the terrestrial irradiation data at the study area concerned. Especially, among the clean energy resources usage, the solar energy utilization gained intensive interest since 1970 principally due to the rising cost of primary energy sources (oil, coal, natural gas). The amount of solar energy received by the surface of the earth per minute is greater than the energy utilization by the entire population in one year. On the other hand, recent issues of global warming, climate change, and greenhouse gas effects are related to the atmospheric pollution due to fossil energy source emissions. Fossil fuel utilization reduction is closely related to recent clean energy sources exploitation and primarily with the use of solar energy.

### 9.2.1 Angström Model

Angström (1924) formulation provides the relationship between the global daily,  $H$ , monthly,  $\bar{H}$ , or yearly,  $\overline{\bar{H}}$ , solar irradiation from the comparatively simple measurements of sunshine duration,  $S$ , by taking into consideration ratios to the extraterrestrial solar irradiation,  $H_0$ , and sunshine duration,  $S_0$ , according to the following simple expression.

$$\frac{H}{H_0} = a + b \frac{S}{S_0} \quad (9.1)$$

Here,  $a$  and  $b$  are model parameters. In general, these parameters are estimated by a simple regression method from simultaneous records of solar irradiation and sunshine duration at a location. The use of regression should take into consideration the satisfaction of the following restrictive assumptions (Sen 2004).

- (1) The model parameters ( $a$  and  $b$ ) are assumed invariant with time on the average as if the same sunshine

- duration appears on the same days or months of the year in a particular location,
- (2) Whatever the scatter diagram of  $H$  versus  $S$ , automatically the regression line is fitted leading to constant  $a$  and  $b$  estimates for the given data. In fact, these coefficients depend on the variations in the sunshine duration during any particular time interval and since sunshine duration records have inherently random variabilities so are the model parameters,
  - (3) Angström approach provides estimations of the global solar irradiation on horizontal surfaces, but unfortunately, it does not give clues about global solar irradiation on a tilted surface, because diffuse and direct irradiations do not appear in the Angström model,
  - (4) Angström linear model relates the global solar irradiation to the sunshine duration only by ignoring the other meteorological factors such as the relative humidity, maximum temperature, air quality, latitude, and elevation above mean sea level. Each one of these factors contributes to the relationship between  $H$  and  $S$ , and their ignorance causes some errors in the prediction and even in the model identification. For instance, Eq. (9.1) assumes that the global solar irradiation on horizontal surfaces is proportional to the sunshine duration only. The effects of other meteorological variables always appear as deviations from the straight-line fit on any scatter diagram all over the world. In order to overcome this, it is necessary to assume that the coefficients in Angström equation are not constants, but rather uncertain variables that may change according to the capacity of measured data (Sect. 9.2.3),
  - (5) The physical meanings of the model coefficients are not considered in most of the application studies, but only the statistical linear regression line fit and parameter estimations are obtained directly and then incorporated into Eq. (9.1) for the global solar irradiation estimation from the sunshine duration records. This is because the

regression method does not provide dynamic estimation of the coefficients from available data.

In general, all or few of these assumptions are not valid for model in Eq. (9.1) of solar irradiation and sunshine duration. In order to calculate  $a$  and  $b$  coefficients in Eq. (9.1), two equations are necessary. These are obtained as the first one by taking the arithmetic average of both sides and the second after multiplication of both sides by the independent term, which is  $H/H_0$ , and then by taking the arithmetic average of this new expression as follows.

$$\left( \frac{\bar{H}}{H_0} \right) = a + b \left( \frac{\bar{S}}{S_0} \right) \quad (9.2)$$

and

$$\left( \frac{\bar{H}}{H_0} \right) \left( \frac{\bar{S}}{S_0} \right) = a \left( \frac{\bar{S}}{S_0} \right) + b \left( \frac{\bar{H}}{H_0} \right)^2 \quad (9.3)$$

The simultaneous solution of these two expressions yields to the following equations in terms of arithmetic averages only.

$$b = \frac{\left( \frac{\bar{H}}{H_0} \right) \left( \frac{\bar{S}}{S_0} \right) - \left( \frac{\bar{H}}{H_0} \right) \left( \frac{\bar{S}}{S_0} \right)}{\left( \frac{\bar{S}}{S_0} \right)^2 - \left( \frac{\bar{S}}{S_0} \right)^2} \quad (9.4)$$

and

$$a = \left( \frac{\bar{H}}{H_0} \right) - b \left( \frac{\bar{S}}{S_0} \right) \quad (9.5)$$

Under the light of these last equations, it is possible to conduct all calculations according to Table 9.1 arrangements. For the parameter estimations, the following table provides a simple way of collective calculation.

Here, the second and third columns are for the sunshine duration and solar radiation data, respectively. The fourth

**Table 9.1** Regression coefficient calculation table

Sample number	Data values		Additional calculations	
	$S/S_0$	$H/H_0$	$(S/S_0)(H/H_0)$	$(H/H_0)^2$
1	$(S/S_0)_1$	$(H/H_0)_1$	$(S/S_0)_1(H/H_0)_1$	$(H/H_0)_1^2$
2	$(S/S_0)_2$	$(H/H_0)_2$	$(S/S_0)_2(H/H_0)_2$	$(H/H_0)_2^2$
3	$(S/S_0)_3$	$(H/H_0)_3$	$(S/S_0)_3(H/H_0)_3$	$(H/H_0)_3^2$
.	.	.	.	.
.	.	.	.	.
$n - 1$	$(S/S_0)_{n-1}$	$(H/H_0)_{n-1}$	$(S/S_0)_{n-1}(H/H_0)_{n-1}$	$(H/H_0)_{n-1}^2$
$n$	$(S/S_0)_n$	$(H/H_0)_n$	$(S/S_0)_n(H/H_0)_n$	$(H/H_0)_n^2$
Average	$\overline{S/S_0}$	$\overline{H/H_0}$	$\overline{(S/S_0)(H/H_0)}$	$\overline{(H/H_0)^2}$

column is for the left-hand side multiplication with right-hand side variable and final column is for the square of the right-hand side variable,  $S/S_0$ . The substitution of the arithmetic averages in the last row in the table into Eqs. (9.4) and (9.5) yields the model parameters.

The following simple MATLAB language written least squares program helps to calculate the model parameters provided that sunshine duration and solar irradiation data are available.

```

function [a,b] = SolarRadiationAngstrom(S,H,Title)
% This program calculates Angström (1924) solar irradiation linear model
% coefficients by the classical regression equation
% S..... Sunshine duration data divided extra-terrestrial sunshine duration value
% H..... Solar irradiance data divided by extra-terrestrial solar irradiation value
% a and b... Angström linear model parameters
AveS=mean(S); % Sunshine duration value arithmetic average
AveH=mean(H); % Solar radiation value arithmetic average
AveSH=mean(S.*H); % Cross multiplication average
SqrS=mean(S.^2); % Square of sunshine duration value
% Coefficient calculations
b=(AveSH-AveS*AveH) / (SqrS-AveS*AveS);
a=mean(S)-b*mean(H);
% Graphical representation
figure
scatter(S,H, 'k*')
% First and last point values
FP=a+b*min(S);
LP=a+b*max(S);
hold on
line([min(S) max(S)], [FP LP], 'LineWidth', 2);
box on
grid on
title>Title)
xlabel('Sunshine duration')
ylabel('Solar irradiation')
legend('Data points','Angström model','Location','SouthEast')
x=min(S)+0.015;
y=max(H)-0.020;
text(x,y,['a = ',num2str(a), ' b = ',num2str(b)])
end

```

The entrance of the following sunshine duration,  $S$ , and solar irradiation,  $H$ , data into the above MATLAB program leads to the scatter of points and the regression line fit as in Fig. 9.1.

Any parameter estimation under the light of these assumptions is referred to as the restrictive model, whereas unrestrictive model is presented in the subsequent section.

### 9.2.2 Unrestrictive Model

To avoid restrictive assumptions, Sen (2001a) presented unrestrictive model concept and its parameter estimation method in the following sequel. Since there are two unknown parameters ( $a$  and  $b$ ) in Eq. (9.1), it is necessary to have two equations for their identification. For this purpose,

the arithmetic average and the variance of Eq. (9.1) are taken leading to the following expressions,

$$\overline{\left(\frac{H}{H_0}\right)} = a' + b' \overline{\left(\frac{S}{S_0}\right)} \quad (9.6)$$

and

$$\text{Var}\left(\overline{H/H_0}\right) = b'^2 \text{Var}\left(\overline{S/S_0}\right) \quad (9.7)$$

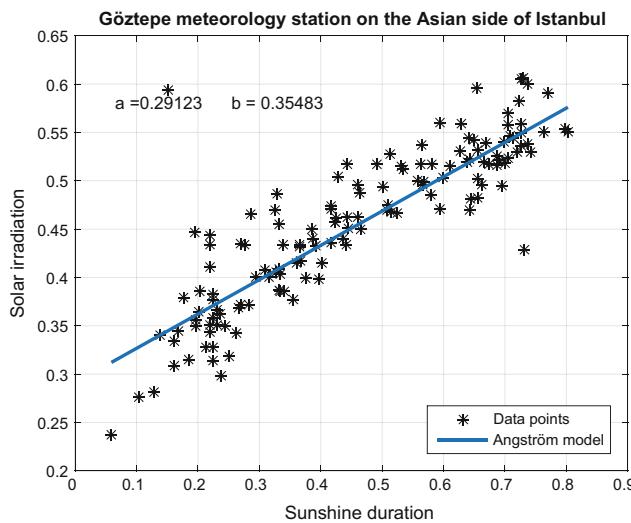
where  $a'$  and  $b'$  are similar parameters as for  $a$  and  $b$  in the Angström expression in Eq. (9.1). The simultaneous solution of Eqs. (9.6) and (9.7) yields parameter estimates as,

$$b' = r_{sh} \sqrt{\frac{\text{Var}(\overline{H/H_0})}{\text{Var}(\overline{S/S_0})}} \quad (9.8)$$

where  $r_{sh}$  implies to the correlation coefficient between the solar irradiation and the sunshine duration. The other parameter estimation can be obtained as,

$$a' = \overline{\left(\frac{H}{H_0}\right)} - \sqrt{\frac{\text{Var}(\overline{H/H_0})}{\text{Var}(\overline{S/S_0})}} \overline{\left(\frac{S}{S_0}\right)} \quad (9.9)$$

Physically, variations in the irradiation data are always smaller than the sunshine data, and consequently,



**Fig. 9.1** Angström model

$\text{Var}(\overline{S/S_0}) \gg \text{Var}(\overline{H/H_0})$ . Equation (9.7) is a special case of classical Angström's coefficient when  $r_{hs} = 1$ , which implies that restrictive assumption bias affects are explainable globally by  $r_{hs}$ .

Mathematically, the second term in Eq. (9.9) is smaller than the first one, and hence,  $a'$  is positive. The following

relationships are valid between the restrictive (Angström equation) and unrestricted model parameters.

$$b' = \frac{b}{r_{hs}} \quad (9.10)$$

and

$$a' = \frac{a}{r_{hs}} - \left(1 - \frac{1}{r_{hs}}\right) \overline{\left(\frac{H}{H_0}\right)} \quad (9.11)$$

These theoretical relationships between the parameters of the two models imply that  $b$  and  $b'$  are the slopes of the straight lines.

Two methods coincide at the centroid point (averages of global solar irradiation and sunshine duration data). Unrestricted model yields overestimates (underestimation) compared to the Angström estimations if sunshine duration data are greater (smaller) than the average value. Equation (9.7) shows that  $a' > a$  and the summation of model parameters is,

$$a' + b' = \frac{a+b}{r_{hs}} - \left(1 - \frac{1}{r_{hs}}\right) \overline{\left(\frac{H}{H_0}\right)} \quad (9.12)$$

The following MATLAB program provides parameter estimations for unrestricted solar energy formulation.

```
function [a,b] = SolarRadiationUnrestricted(S,H,Title)
% This program calculates Şen(2001) restricted solar irradiation linear model
% coefficients by the classical regression equation
% S..... Sunshine duration data divided extra-terrestrial sunshine duration value
% H..... Solar irradiance data divided by extra-terrestrial solar irradiation value
% a and b... Şen linear model parameters
AveS=mean(S); % Sunshine duration value arithmetic average
AveH=mean(H); % Solar radiation value arithmetic average
VarS=var(S); % Sunshine duration value variance
VarH=var(H); % Solar radiation value arithmetic average
AveSH=mean(S.*H); % Cross multiplication average
SqrS=mean(S.^2); % Square of sunshine duration value
% Coefficient calculations
b=sqrt(VarH/VarS);
a=AveH-b*AveS;
% Graphical representation
figure
scatter(S,H,'k*')
% First and last point values
FP=a+b*min(S);
LP=a+b*max(S);
hold on
line([min(S) max(S)], [FP LP], 'LineWidth',2);
box on
grid on
title>Title)
xlabel('Sunshine duration')
ylabel('Solar irradiation')
legend('Data points','Şen model','Location','SouthEast')
x=min(S)+0.015;
y=max(H)-0.020;
text(x,y,['a =',num2str(a), ' b = ',num2str(b)])
end
```

Figure 9.2 is the output of the above MATLAB program for the restrictive model with data values in Table 9.2. According to Eq. (9.10), the Angström (restricted) equation slope is smaller than the unrestricted approach ( $b < b'$ ), since always  $r_{hs} > 0$  for global solar irradiation and sunshine duration data scatter on a Cartesian coordinate system (see Fig. 9.2).

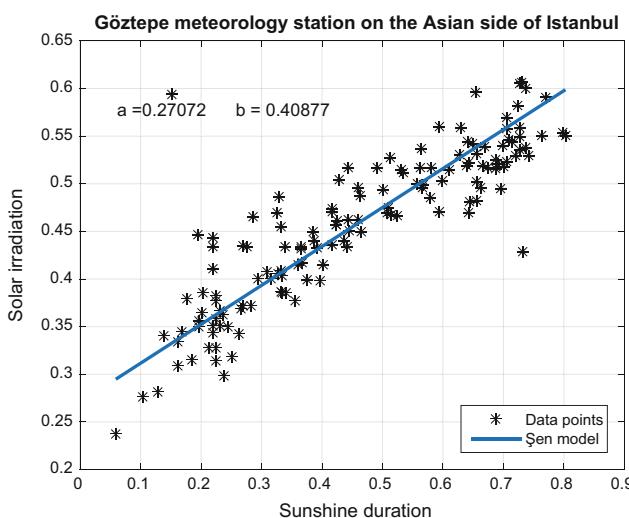
In order to appreciate the difference between the Angström and Şen (unrestrictive) methods they are drawn on the same graph as in Fig. 9.3.

One can clearly see that the classical Angström model slightly overestimates solar irradiation values more than the Şen model for sunshine duration values less than the arithmetic average, whereas just the opposite is valid for sunshine duration values bigger than the average.

### 9.2.3 Successive Substitution Model

As already mentioned in the previous sections, solar energy expressions have two parameters,  $a$  and  $b$ , which are estimated from available sunshine duration and solar irradiation records by statistical methodologies. All the estimation methods yield single value for these parameters, and hence, local deviations (unexplained parts) from the final straight-line model are exterminated without any value. In order to consider effects of unexplained parts, it is necessary to estimate coefficients from the successive data pairs “locally” rather than “globally” as in the other approaches.

Let us first consider the physical and mathematical meanings of parameters  $a$  and  $b$  in Eq. (9.1). First of all  $a$  represents the ratio of actual daily global irradiation,  $H$ , to the daily extraterrestrial irradiation,  $H_0$ , provided that



**Fig. 9.2** Restricted solar energy Şen linear model

physically the sun is covered by clouds all day, so that  $S = 0$ , i.e., over-casted sky. On the other hand,  $b$  corresponds to the slope of the linear relationship which is defined as,

$$b = \frac{d(H/H_0)}{d(S/S_0)} \quad (9.13)$$

This first-order ordinary differential equation can be written in terms of backward finite difference method as,

$$b'_i = \frac{\left(\frac{H}{H_0}\right)_i - \left(\frac{H}{H_0}\right)_{i-1}}{\left(\frac{S}{S_0}\right)_i - \left(\frac{S}{S_0}\right)_{i-1}} \quad (i = 2, 3, \dots, n) \quad (9.14)$$

Herein,  $n$  is the number of records and  $b'_i$  is the rate of global irradiation change with the sunshine duration between time instances,  $i - 1$  and  $i$ . For daily data, these are successive daily rates of change or in the case of monthly records, monthly rates of change. Arrangement from Eq. (9.1) by considering Eq. (9.14) leads to the successive time estimates of  $a'_i$  as,

$$a'_i = \left(\frac{H}{H_0}\right)_i - b'_i \left(\frac{S}{S_0}\right)_i \quad (i = 2, 3, \dots, n) \quad (9.15)$$

The application of these last assumption free equations yields  $(n - 1)$  successive coefficient estimations at lag = 1. Each pair of the coefficient estimate  $(a'_i, b'_i)$  explains all local unexplained part inclusive information for successive pairs of global radiation and corresponding sunshine duration records. On the other hand, estimations of Angström's coefficients by the application of regression technique are obtained as single values,

$$b = \frac{\sum_{i=1}^n \left[ \left(\frac{H}{H_0}\right)_i - \overline{\left(\frac{S}{S_0}\right)_i} \right] \left[ \left(\frac{H}{H_0}\right)_{i-1} - \overline{\left(\frac{S}{S_0}\right)_{i-1}} \right]}{\sqrt{\sum_{i=1}^n \left[ \left(\frac{H}{H_0}\right)_i - \overline{\left(\frac{S}{S_0}\right)_i} \right]^2 \left[ \left(\frac{H}{H_0}\right)_{i-1} - \overline{\left(\frac{S}{S_0}\right)_{i-1}} \right]^2}} \quad (9.16)$$

and

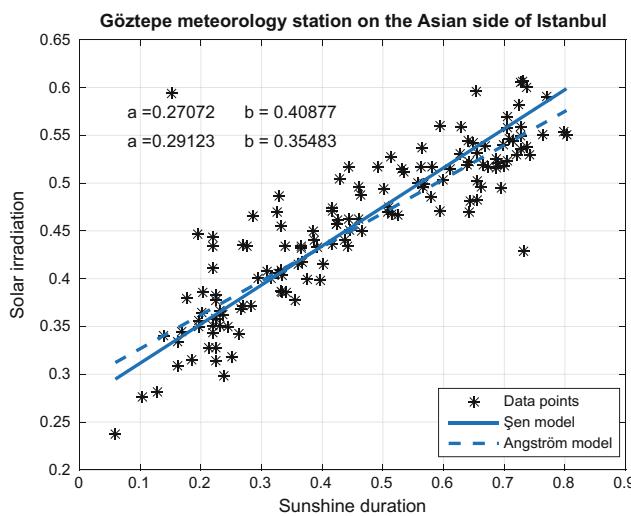
$$a = \overline{\left(\frac{H}{H_0}\right)} - b \overline{\left(\frac{S}{S_0}\right)} \quad (9.17)$$

Comparisons of Eq. (9.14) with Eqs. (9.16) and (9.15) with (9.17) indicate that the regression technique estimations do not allow any randomness in the coefficients calculations. The finite difference method coefficient estimations do not depend on the regression technique assumptions. This provides flexibility in the parameter estimation.

Furthermore, it is possible to obtain the relative frequency distribution of Angström coefficients,  $a'_i$  and  $b'_i$ . Additionally, one can also calculate any statistical parameter such as the variance or standard deviation of each parameter. Last

**Table 9.2** Istanbul City Asian side data

$S$	$H$	$S$	$H$	$S$	$H$	$S$	$H$	$S$	$H$	$S$	$H$	$S$	$H$
0.25	0.32	0.42	0.46	0.80	0.55	0.40	0.40	0.34	0.39	0.53	0.52	0.58	0.52
0.21	0.33	0.16	0.31	0.71	0.54	0.57	0.50	0.44	0.46	0.38	0.45	0.77	0.59
0.28	0.37	0.29	0.40	0.72	0.53	0.65	0.60	0.38	0.40	0.33	0.49	0.73	0.43
0.33	0.40	0.24	0.30	0.42	0.46	0.73	0.54	0.50	0.49	0.22	0.38	0.49	0.52
0.44	0.44	0.10	0.28	0.27	0.43	0.70	0.49	0.45	0.45	0.20	0.36	0.28	0.43
0.71	0.52	0.19	0.32	0.29	0.47	0.74	0.54	0.57	0.50	0.20	0.35	0.19	0.45
0.66	0.50	0.27	0.37	0.22	0.36	0.33	0.41	0.67	0.52	0.24	0.35	0.33	0.39
0.64	0.52	0.52	0.47	0.20	0.36	0.22	0.38	0.69	0.52	0.39	0.44	0.22	0.34
0.74	0.53	0.66	0.50	0.26	0.34	0.18	0.38	0.64	0.52	0.64	0.54	0.37	0.43
0.51	0.47	0.64	0.48	0.56	0.54	0.16	0.33	0.42	0.44	0.71	0.56	0.46	0.46
0.44	0.43	0.64	0.47	0.59	0.56	0.22	0.35	0.27	0.37	0.69	0.52	0.37	0.42
0.22	0.44	0.80	0.55	0.72	0.58	0.39	0.43	0.43	0.50	0.56	0.52	0.71	0.57
0.22	0.31	0.61	0.51	0.73	0.61	0.32	0.40	0.33	0.41	0.42	0.47	0.66	0.48
0.23	0.35	0.33	0.47	0.73	0.61	0.46	0.49	0.53	0.51	0.33	0.46	0.71	0.55
0.31	0.41	0.22	0.43	0.63	0.56	0.56	0.50	0.63	0.53	0.15	0.59	0.74	0.60
0.46	0.45	0.13	0.28	0.42	0.47	0.73	0.55	0.42	0.46	0.17	0.34		
0.51	0.48	0.14	0.34	0.34	0.43	0.76	0.55	0.51	0.47	0.23	0.37		
0.58	0.49	0.35	0.38	0.20	0.39	0.70	0.52	0.70	0.54	0.36	0.42		
0.59	0.47	0.40	0.42	0.06	0.24	0.33	0.39	0.67	0.54	0.46	0.50		
0.69	0.52	0.60	0.50	0.24	0.36	0.22	0.33	0.73	0.56	0.51	0.53		
0.65	0.54	0.67	0.52	0.37	0.43	0.22	0.41	0.65	0.53	0.44	0.52		

**Fig. 9.3** Angström and Sen linear models

but not least, confidence limits can be stated at a certain significance level as 5 or 10%, for each coefficient's estimation. Extreme values of  $a'_i$  and  $b'_i$  become observable by the finite difference method solution.

Taking the average value of both sides in Eq. (9.15) leads to finite difference averages of the new Angström coefficients as,

$$\bar{a}' = \overline{\left(\frac{H}{H_0}\right)} - b'_i \overline{\left(\frac{S}{S_0}\right)} \quad (9.18)$$

The difference of this expression from Eq. (9.17) results as,

$$\bar{a}' - a = (b - \bar{b}') \overline{\left(\frac{S}{S_0}\right)} \quad (9.19)$$

All what have been explained in this section can be obtained by the execution of the following MATLAB program.

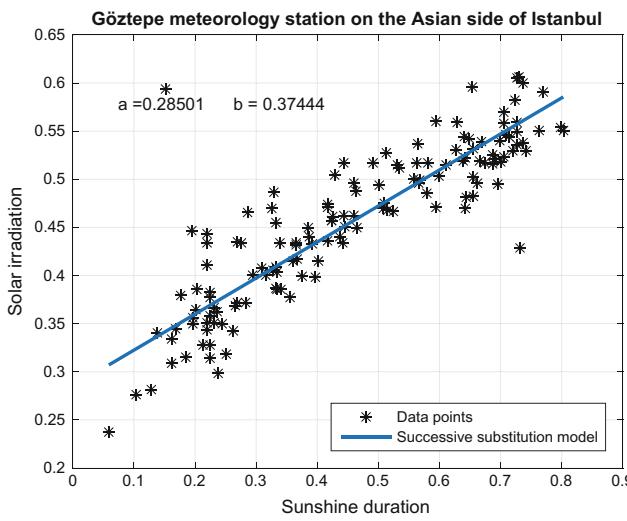
```

function [a,b] = SolarRadiationSuccessiveSubstitution(S,H,Title)
% This program calculates Angström (1924) solar irradiation linear model
% coefficients by the classical regression equation
% S..... Sunshine duration data divided extra-terrestrial sunshine duration value
% H..... Solar irradiance data divided by extra-terrestrial solar irradiation value
% a and b... Angström linear model parameters
% Coefficient calculations
N=10;
k=0;
for i=1:N-1
    i1=i+1;
    for j=i1:N
        k=k+1;
        b(k)=(H(i)-H(j))/(S(i)-S(j));
        a(k)=H(i)-b(k)*S(i);
    end
end
Avea=mean(a);
Aveb=mean(b);
% Graphical representation
figure
scatter(S,H,'k*')
% First and last point values
FP=Avea+Aveb*min(S);
LP=Avea+Aveb*max(S);
hold on
line([min(S) max(S)], [FP LP], 'LineWidth', 2);
box on
grid on
title>Title)
xlabel('Sunshine duration')
ylabel('Solar irradiation')
legend('Data points','Successive substitution model','Location','SouthEast')
x=min(S)+0.015;
y=max(H)-0.020;
text(x,y,['a = ',num2str(Avea), ' b = ',num2str(Aveb)])
end

```

Figure 9.4 results from the running of above MATLAB language program with data given in Table 9.2.

Figure 9.5 provides opportunity to compare the previous three linear solar energy models.

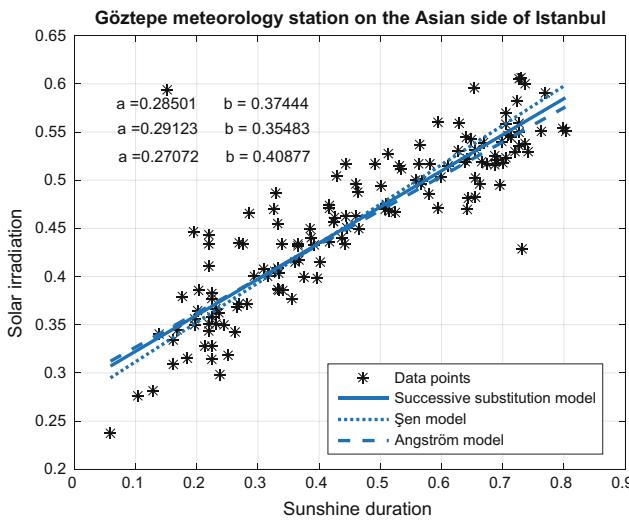


**Fig. 9.4** Successive substitution linear models

#### 9.2.4 Nonlinear Models

Global solar irradiation and sunshine duration data have unimodal unsymmetrical probability distribution function (PDF). This may be due to the areal and depth cloud formations with PDFs, which approximately follow a power-law (Davis 1978). More intense cloud areas are found to have comparatively bigger powers as indicative of the increase in turbulence at the cloud top and decrease in the global solar irradiation. A number of theoretical studies have shown the sensitivity of cloud radiative properties to their spatial structure, which, in turn, affects the sunshine duration and subsequently the global irradiation amounts (Joseph and Cahalan 1988). It is, therefore, very significant to preserve in any estimation procedure the third or higher statistical moments (Sen 2007). These ideas bring into one's mind the possibility of sunshine duration variable third power term similar to reference (Akinoğlu and Ecevit 1990).

$$\frac{H}{\bar{H}_0} = a + b \frac{S}{\bar{S}_0} + c \left( \frac{S}{\bar{S}_0} \right)^2 + d \left( \frac{S}{\bar{S}_0} \right)^3 \quad (9.20)$$



**Fig. 9.5** Comparison of the three linear solar energy models

where  $c$  and  $d$  are additional model parameters. There are some mathematical and physical implicative assumptions in this model (Şen 2007).

- (1) In any modeling approach, parsimony (i.e., to obtain the best result with least number of parameters) is one of the desirable requirements,
- (2) It is difficult to explain on physical grounds why a polynomial expression is adopted for the modeling purposes apart from the fact that these expressions have mathematical convenience,
- (3) In statistical literature, second-order statistics (variance) subsumes first-order statistics (average) and third-order statistics (skewness) includes first- and second-order statistics (Caughey et al. 1982). In general, a polynomial model leads to imbedded redundancy in the model.

Under the light of these points, the following power model is suggested for more flexible solar irradiation estimations,

$$\frac{H}{H_0} = a + b \left( \frac{S}{S_0} \right)^c \quad (9.21)$$

where  $c$  is the parameter that represents relative diffuse radiation on any overcast day and the summation of  $a$  and  $b$  represents the relative cloudless-sky global irradiation. Average solar irradiation,  $\overline{(H/H_0)}$ , and sunshine duration,  $\overline{(S/S_0)}$ , values renders Eq. (9.21), into,

$$\left( \frac{\overline{H}}{\overline{H}_0} \right) = a + b \left( \frac{\overline{S}}{\overline{S}_0} \right)^{1/c} \quad (9.22)$$

If the power in Eq. (9.22) turns out as approximately equal to 1, then in solar energy calculations Angström equation can be used. As in Fig. 9.1 the Angström equation passes through the scatter diagram center, the power model expression does not pass from this center. In Fig. 9.6 different power parameter values are classified into groups, as for  $c > 1$ , the convex nonlinear power models; linear Angström equation for  $c = 1$ , and finally, concave nonlinear power models for  $c < 1$ .

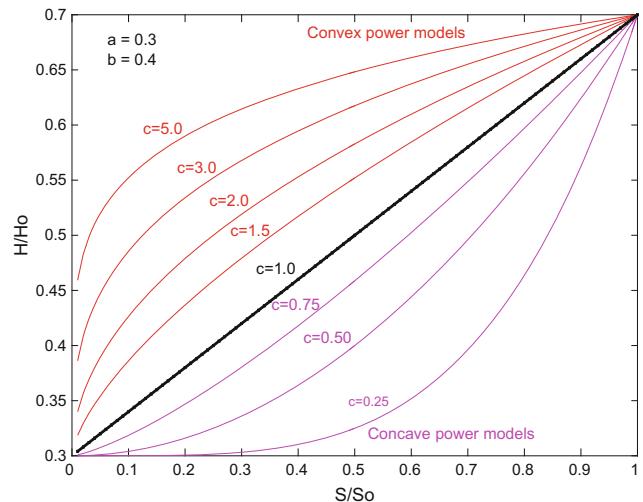
At the extreme weather conditions such as overcast ( $S/S_0 = 0$ ) and clear-sky ( $S/S_0 = 1$ ) situations, there may not appear appreciable difference between the power and classical Angström approaches. Real divergence between the two models appears at the intermediate sunshine duration and solar irradiation data values.

Since Eq. (9.22) has a nonlinear form, the parameter estimations can be obtained by the nonlinear least squares analysis, which minimizes the prediction error sum of squares (SS). The partial derivatives of the SS with respect to each model parameter should be equal to zero. The necessary formulations of the nonlinear least squares analysis are given by Şen (2008). The two basic parameter estimation formulations are as follows.

$$a = \frac{1}{n} \sum_{i=1}^n \overline{\left( \frac{H}{H_0} \right)_i} - b \frac{1}{n} \sum_{i=1}^n \overline{\left( \frac{S}{S_0} \right)_i}^{1/c} \quad (9.23)$$

and,

$$b = \frac{\frac{1}{n} \sum_{i=1}^n \left( \overline{\frac{H}{H_0}} \right)_i \left( \overline{\frac{S}{S_0}} \right)_i^{1/c} - \left[ \frac{1}{n} \sum_{i=1}^n \left( \overline{\frac{H}{H_0}} \right)_{i-1} \right] \left[ \frac{1}{n} \sum_{i=1}^n \left( \overline{\frac{S}{S_0}} \right)_i^{1/n} \right]}{\frac{1}{n} \sum_{i=1}^n \left( \overline{\frac{S}{S_0}} \right)_i^{2/c} - \left[ \frac{1}{n} \sum_{i=1}^n \left( \overline{\frac{S}{S_0}} \right)_i^{1/c} \right]^2} \quad (9.24)$$



**Fig. 9.6** Different power model alternatives

For  $c = 1$ , these two equations reduce to the classical regression line parameter estimates for the Angström model in Sect. 9.2.1. Analytical solution of the three parameters simultaneously is not possible, and therefore, a numerical optimization procedure is used. The necessary steps of this procedure are as follows.

- (1) A small initial  $c$  value is chosen and with the solar irradiation and sunshine duration data at hand, corresponding  $a$  and  $b$  parameter estimates are calculated from Eqs. (9.23) and (9.24), respectively,

- (4) When  $s_{j+1}$  becomes less than a preselected critical value,  $s_{cr}$ , then the optimization procedure is stopped and the model parameter estimates are taken corresponding to this situation.

The steps (2)–(4) are repeated until the optimum estimation point is reached according to the adopted critical value. In practice,  $s_{cr}$  can be taken as 0.01 or 0.05 depending on the desired accuracy level.

Power solar calculations can be achieved by the following MATLAB language written program.

```
function [a, b, ss] = powersolrad(s,h,c)
% This program accepts s = S/S0 and h = H/H0 values and
% optimizes a and b by changing c,
% and finally arrives at these three parameter estimates
% with the least sum of square error.
% n is the number of data,
% ss is the sum of squares
n=length(s);
cu=1/c;
cu2=2*cu;
sd=s.^cu;
sdh=sd.*h;
aves = sum(sd)/n;
aveh = sum(h)/n;
avesqs = sum(s.^cu2)/n;
avesh = sum(sdh)/n;
b = (avesh - aveh*aves) / (avesqs - aves*aves);
a = aveh - b*aves;
ss = 0;
for i=1:n
    ss= ss + (h(i) - a - b*s(i).^cu)^2;
end
```

- (2) With this set of parameter estimates the sum of squares (SS) errors value is calculated from the following expression (Sen 2008),

$$SS = \sum_{i=1}^n \overline{\left(\frac{H}{H_o}\right)} - a - b \left(\overline{\frac{S}{S_o}}\right)_i^{1/n} \quad (9.25)$$

- (3) The slope,  $s_{j+1}$ , value of SS variation with  $c$  value is calculated by taking the difference between two successive SS values divided by the increment  $\Delta c$  in  $c$  as,

$$s_{j+1} = \frac{(SS)_{j+a} - (SS)_j}{\Delta c} \quad (9.26)$$

Here, the counter  $j$  shows the previous step calculation in the numerical optimization procedure.

## 9.2.5 Probabilistic Innovative Solar Energy Estimation

All previous solar irradiation estimation model parameters are determined by the statistical and mathematical procedures. In this section, an innovative probability methodology (IPM) is proposed as suggested by Sen (2016) for solar irradiation estimation based on the cumulative probability distribution functions (CDF) of the sunshine duration and solar irradiation data. The basis of the methodology is to overlap by probabilistic transformation the CDFs of solar irradiation and sunshine duration data.

Figure 9.7 indicates the methodological steps of the innovative probabilistic solar irradiation estimation.

In the following, MATLAB program steps are explained for the successive completion of the proposed methodology (Sen 2016).

- (a) Determine the plotting position of  $S/S_0$  data, by sorting the data into ascending order. Attach to each data value a probability,  $P_m$ , of exceedence according to the following expression,

$$P_m = \frac{m}{n + 1} \quad (9.27)$$

where  $m$  is the order and  $n$  is the number of data. The corresponding MATLAB statements are

```
n=length(SSo);
Pm=(1:1:n)/(n+1);
```

- (b) the best CDF is fitted to sorted data among the normal (Gaussian), logarithmic-normal, Gamma, Weibull, and extreme value (Gumbel 1958) distributions. Detailed information about these CDFs can be found in Benjamin and Cornell (1970). The mathematical functions of these CDFs are given in sequence as follows each with MATLAB program statements by considering that SS<sub>0</sub> as the vector of sunshine duration standard data.

Normal PDF MATLAB statements are,

```
par=normfit(SSo);
fx=normcdf(SSo,par(1),par(2));
```

Log-normal PDF MATLAB statements are,

```
par=lognfit(SSo);
fx=logncdf(SSo,par(1),par(2));
```

Gamma PDF MATLAB statements are,

```
par=gamfit(SSo);
fx=gamcdf(SSo,par(1),par(2));
```

Weibull PDF MATLAB statements are,

```
par=wblfit(SSo);
fx=wblcdf(SSo,par(1),par(2));
```

Extreme value PDF MATLAB statements are,

```
par=evfit(SSo);
fx=evcdf(SSo,par(1),par(2));
```

The most suitable CDF should have the minimum sum of square error from the theoretical CDS.

- (c) Repeat steps *a* and *b* for the solar irradiation data,  $H/H_0$ . It is not necessary that the sunshine duration and solar irradiation data should have a similar type of CDFs,
- (d) The next step is to convert the sunshine duration CDF to solar irradiation CDF. In order to do this, the  $S/S_0$  data are entered to the sunshine CDF and the corresponding theoretical probability set is calculated, and then these probabilities are entered into the solar radiation CDF, which results in a new set of converted  $H/H_0$  values independent from the given data set of  $H/H_0$ ,
- (e) The new set of solar irradiation values is in good relationship with the measurements, because after all they are coming from the same CDF. The relationship is determined between  $H_c/H_0$  and  $H/H_0$  values,
- (f) After the completion of all the previous steps, now there are CDFs of sunshine duration and solar irradiation variables, and final relationship between the measured  $H/H_0$  and converted  $H_c/H_0$  one can make prediction,  $H_p/H_0$ , for any given  $S/S_0$  value as shown in Fig. 9.7.

The following MATLAB language written programs help to complete the necessary calculations for the IPM approach to solar energy records for modeling.

```

function [FMOP] = ProbabilisticSystemModeling(X,Y,ITitle, OTitle)
% This program performs probabilistic fuzzy system modeling with a number
% of input CDF and one output CDF
% The program is written by Zekai Sen on 31 October 2015 Saturday at 5.05
% PM
% X      : Input variable vectors:
% Y      : Output vector data
% Y      : Model output data series
% SX     : Sorted data series
% ITitle : Input variable titles
% OTitle : Output variable title
% CDFy   : Theoretical CDF ordinates (0 < CDFy < 1)
% CDFpram : Theoretical CDF parameters
% CDFTitle: Theoretically suitable CDF to given data
% FMOP    : Model output prediction
% Various CDF considered in the program
%           I = 1, Normal CDF
%           I = 2, Logarithmic-normal CDF
%           I = 3, Exponential CDF
%           I = 4, Gamma CDF
%           I = 5, Weibull CDF
%           I = 6, Gumbel (Extreme value) CDF
n=size(X);
ND=n(1,1);      % Number of data in the variables
NI=n(1,2);      % Number of input variables
% INPUT DATA EMPIRICAL AND THEORETICAL CDFs
for j=1:NI % Number of variables
    [II,SXX,CDFyy,CDFpramm,CDFTitlee] = CDFit(X(:,j),ITitle);
    I(j)=II;
    SX(j,1:ND)=SXX;
    CDFy(j,1:ND)=CDFyy;
    if I(j)==3
        CDFpram(j,1:1)=CDFpramm; % One parameter PDF
    else
        CDFpram(j,1:2)=CDFpramm; % Two parameter PDFs
    end
    % CDFTitle(j,1:ID)=CDFTitlee;
end
% OUTPUT VARIABLE PROBABILITY VALUE CALCULATION
[IO,SY,CDFyO,CDFpramm,CDFTitlee] = CDFit(Y,ITitle);
CDFOprom=CDFpramm;
% INPUT VARIABLES PROBABILITY VALUE CALCULATIONS
for j=1:NI % Number of variables
    if I(j)==1
        P=normcdf(X(:,j),CDFpram(j,1),CDFpram(j,2));
    elseif I(j)==2
        P=logncdf(X(:,j),CDFpram(j,1),CDFpram(j,2));
    elseif I(j)==3
        P=expcdf(X(:,j),CDFpram(j,1));
    elseif I(j)==4
        P=gamcdf(X(:,j),CDFpram(j,1),CDFpram(j,2));
    elseif I(j)==5
        P=wblcdf(X(:,j),CDFpram(j,1),CDFpram(j,2));
    elseif I(j)==6
        P=evcdf(X(:,j),CDFpram(j,1),CDFpram(j,2));
    else
    end
    prob(j,1:ND)=P;
end
for i=1:ND % Number of data in the variables
    OutputProb(i)=min(prob(:,i));
end
% OUTPUT VARIABLE VALUES CALCULATION
for i=1:ND % Number of data in the output variable
    if IO==1
        V=normcdf(OutputProb(i),CDFOprom(1),CDFOprom(2));
    elseif IO==2
        .....Continued.....

```

```

V=logninv(OutputProb(i),CDFOpram(1),CDFOpram(2));
elseif IO==3
    V=expinv(OutputProb(i),CDFOpram(1));
elseif IO==4
    V=gaminv(OutputProb(i),CDFOpram(1),CDFOpram(2));
elseif IO==5
    V=wblinv(OutputProb(i),CDFOpram(1),CDFOpram(2));
elseif IO==6
    V=evinv(OutputProb(i),CDFOpram(1),CDFOpram(2));
else
end
MOP(i)=V;
end
[IMOP,SMOP,CDFyMOP,CDFpramMOP,CDFTitleMOP] = CDFit(MOP,ITitle);

% FINAL MODEL OUTPUT PREDICTION CALCULATIONS BY CDF CONVERSION
% A) FIRST CALCULATE PROBABILITIES ACCORDING TO MOP OUTPUT
if IMOP==1
    pr=normcdf(MOP,mean(MOP),std(MOP));
    FMOP=norminv(pr,CDFOpram(1),CDFOpram(2));
elseif IMOP==2
    pr=logncdf(MOP,CDFpramMOP(1),CDFpramMOP(2));
    FMOP=logninv(pr,CDFOpram(1),CDFOpram(2));
elseif MOP==3
    pr=expcdf(MOP,CDFpramMOP(1));
    FMOP=expinv(pr,CDFOpram(1));
elseif IMOP==4
    pr=gamcdf(MOP,CDFpramMOP(1),CDFpramMOP(2));
    FMOP=gaminv(pr,CDFOpram(1),CDFOpram(2));
elseif IMOP==5
    pr=wblcdf(MOP,CDFpramMOP(1),CDFpramMOP(2));
    FMOP=wblinv(pr,CDFOpram(1),CDFOpram(2));
else
    pr=evcdf(MOP,CDFpramMOP(1),CDFpramMOP(2));
    FMOP=evinv(pr,CDFOpram(1),CDFOpram(2));
end
% THEN CALCULATE MODEL OUTPUT PREDICTIONS ACCORDING TO OBSERVATION OUTPUT
if IO==1
    pr=normcdf(MOP,mean(MOP),std(MOP));
    FMOP=norminv(pr,CDFOpram(1),CDFOpram(2));
elseif IO==2
    FMOP=logninv(pr,CDFOpram(1),CDFOpram(2));
elseif IO==3
    FMOP=expinv(pr,CDFOpram(1));
elseif IO==4
    FMOP=gaminv(pr,CDFOpram(1),CDFOpram(2));
elseif IO==5
    FMOP=wblinv(pr,CDFOpram(1),CDFOpram(2));
else
    FMOP=evinv(pr,CDFOpram(1),CDFOpram(2));
end
pO=(1:1:ND)/(ND+1);
figure
scatter(SY,pO,'k*') % Observation data empirical PDF scatter diagram
hold on
xlabel(OTitle)
ylabel('exceedence probability')
grid on
box on
Ymin=min(Y);
Ymax=max(Y);
scatter(sort(FMOP),pO,'kv') % Model output prediction empirical CDF scatter diagram
legend('Observation output data','Generation output data')
figure
scatter(Y,FMOP,'k*')
grid on
box on
.....Continued.....

```

```

Pmin=min(FMOP);
Pmax=max(FMOP);
Min=min(Ymin,Pmin);
Max=max(Ymax,Pmax);
line([Min Max],[Min Max], 'LineWidth', 2, 'Color', 'k')
 xlabel('Observation data')
 ylabel('Prediction data')
end

```

### 9.2.5.1 Applications

The application is presented for Adana meteorology station in Turkey as a representative of the Mediterranean climatic region. The above MATLAB program plots the data according to Eq. (9.27) and superimposes the best CDF based on the least squares technique. For Adana station the plotting positions with the theoretical CDFs are presented in Fig. 9.8.

Both the sunshine duration and solar irradiation data confirm with Weibull CDF, but with different parameter values. It is necessary to convert the sunshine duration CDF to solar irradiation as shown in Fig. 9.8. This procedure leads to Fig. 9.9, where measurement  $H/H_0$  and converted  $S/S_0$  follow each other closely along the Weibull CDF as in Fig. 9.8b.

This figure provides a common basis for derivation of converted solar irradiation,  $H_c/H_0$ , values as explained in Fig. 9.8. After the calculation of  $H_c/H_0$  values from this CDF, it is time to look for the relationship between these values and the measurement solar irradiation data,  $H/H_0$ . Since  $H_c/H_0$  and  $H/H_0$  are now from the same CDF, there is a good relationship expectation between these two sets of values. Figure 9.10 presents this relationship with the least squares regression line as,

$$\frac{H_c}{H_0} = 0.1862 + 0.5155 \frac{S_c}{S_0} \quad (9.28)$$

In order to show over performance of the proposed IPM, the results are compared with the classical Angström (1924) and successive substitution (Şahin and Şen 1998) solar irradiation prediction methods. These two methods are explained in detail in a textbook by Şen (2008). Figure 9.11 indicates the solar irradiation estimation values change with data number concerning each method.

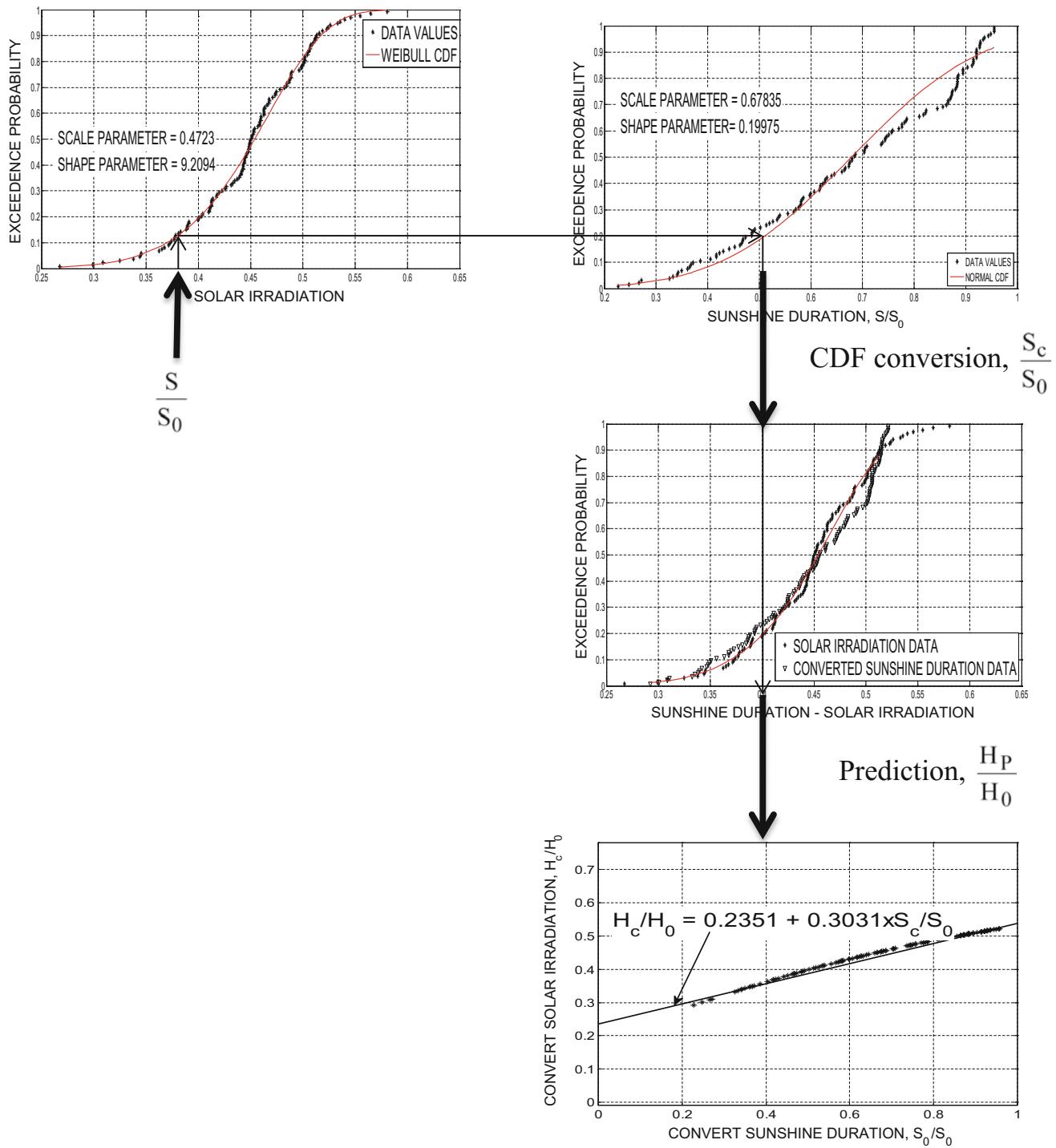
In the same figure, the sum of squares of the deviations from the measured data is calculated for each method and it is observed that the least sum of squares is with the IPM. Comparison of the sum of squares indicates that the suggested method has threefold–fourfold less value. Furthermore, to support the validity of the IPM another graph is prepared in Fig. 9.12, where each method predictions are plotted versus the measurement data on the horizontal axis.

In this figure, the 1:1 ( $45^\circ$ ) straight-line indicates the case of perfect match between the model results and the measurements. It is obvious that the classical models are biased in their predictions, because they do not yield results parallel to the perfect line. The IPM yields scatter of points around the perfect line which is a desired result for a model to be representative of the given data.

## 9.3 Wind Energy

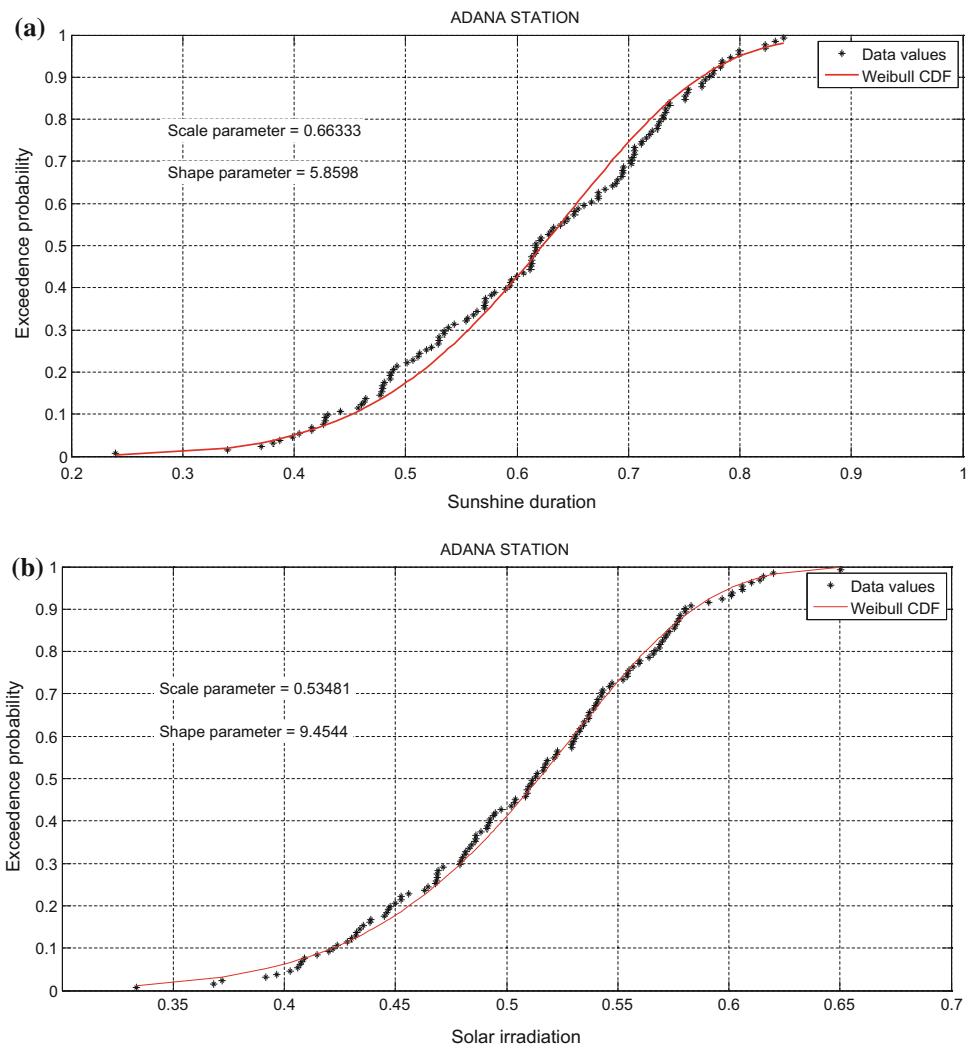
Wind as a meteorological variable has been exploited as energy source by human beings since ancient civilizations. Initially, wind power has been used for water withdrawal by farmers, but later sailors used it for navigation purposes. This source of energy has become an inspiration source for creative activities of human beings, and it has been used also for grinding wheat and other dry agricultural products. Such inspirations are visible even today in the mythological aspects, poems, and songs. Eastern civilizations have introduced the wind power to the Western societies, but its significance has been lost until middle of the twentieth century. Thermic centers have been comparatively very preferred that worked with the oil, coal, natural gas, and other fossil energy sources. It is possible to generate energy by these sources at any place and time easily compared to the wind power. However, for a long time, non-sustainability and non-friendly nature of these fossil fuels have not been understood. On the other hand, the energy crisis during 1970s have given the necessary alert to the world as for the use of renewable and clean energy resources in addition to concise environmental protection. Hence, the wind energy has gained its historical significance again as renewable and clean energy source. Due to spatial and temporal heating of the earth's surface, the winds are generated and various wind energy formulations are available.

The most significant point in the wind energy generation is that the data should be reliable. This is due to the fact that since the wind power is directly proportional to the cube of wind speed, any high or low wind speed measurement will cause important differences in the energy calculations.

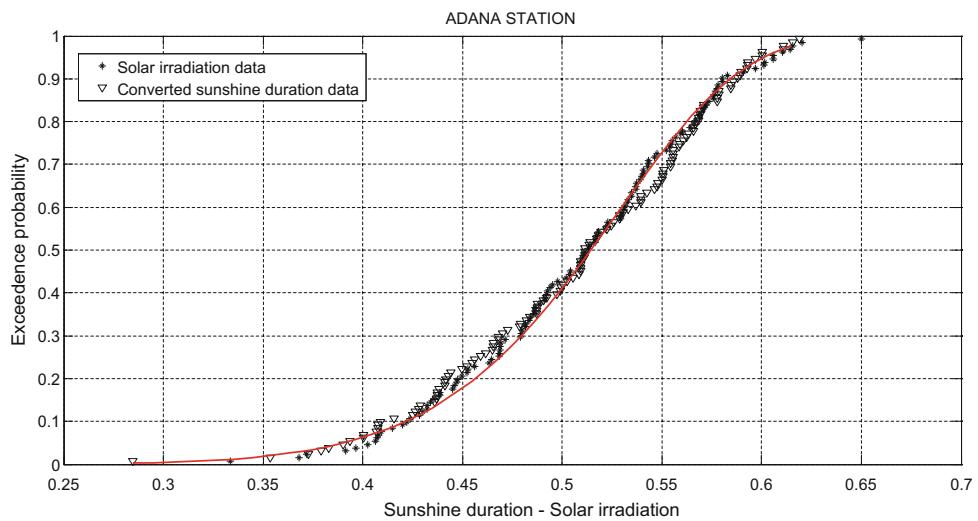


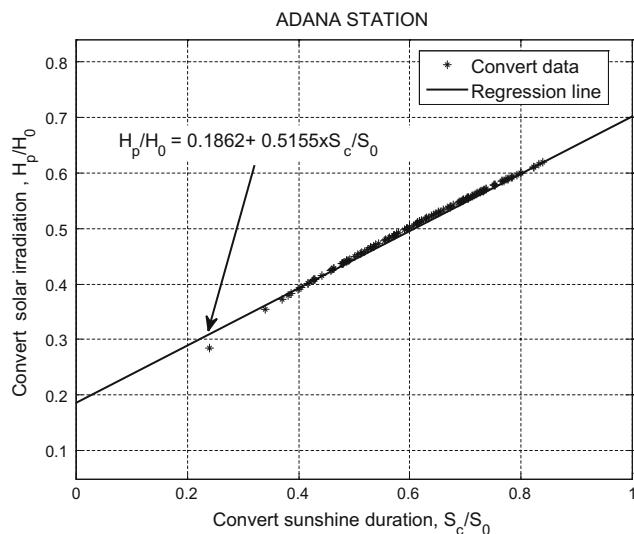
**Fig. 9.7** Innovative probabilistic solar irradiation prediction chart

**Fig. 9.8** Adana station empirical and theoretical CDFs, **a** Sunshine duration, **b** solar irradiation



**Fig. 9.9**  $H/H_0$  and converted  $S/S_0$  empirical and theoretical CDFs





**Fig. 9.10** Relationship between  $H/H_0$  and  $H_c/H_0$

### 9.3.1 Topographic Features

Dynamic meteorological structures in the troposphere and the topographic features cause temporal and spatial variabilities in wind speed. In large-scale wind energy utilization establishments' feasibility depends on the areal wind variability in addition to siting, sizing, operation, and maintenance policies (Sen 1999, 2001b).

On area basis, in order to predict the winds at one site from records at others, it is necessary to have detailed information on terrain and weather patterns. Although the cross-correlation function definition can give a direct indication of the dependence of variations from the mean at any two sites, it suffers from the following drawbacks.

- (i) Autocorrelation and cross-correlation formulations require symmetrically normal (Gaussian) frequency

distribution of wind speed data for reliable calculations. It is well established in the literature that wind speeds accord mostly with a Weibull, Gamma, or logarithmic-normal PDFs, which are all skewed,

- (ii) The correlation function measures the variation around the arithmetic average values of the wind speed data at individual sites. However, in the regional calculations, a measure of relative variability between two sites is necessary. For this purpose, the semi-variogram (SV) or cumulative semi-variogram (CSV) concepts are developed and their modification as the PCSV is presented and used for the regional assessment of the wind variability (see Chap. 8).

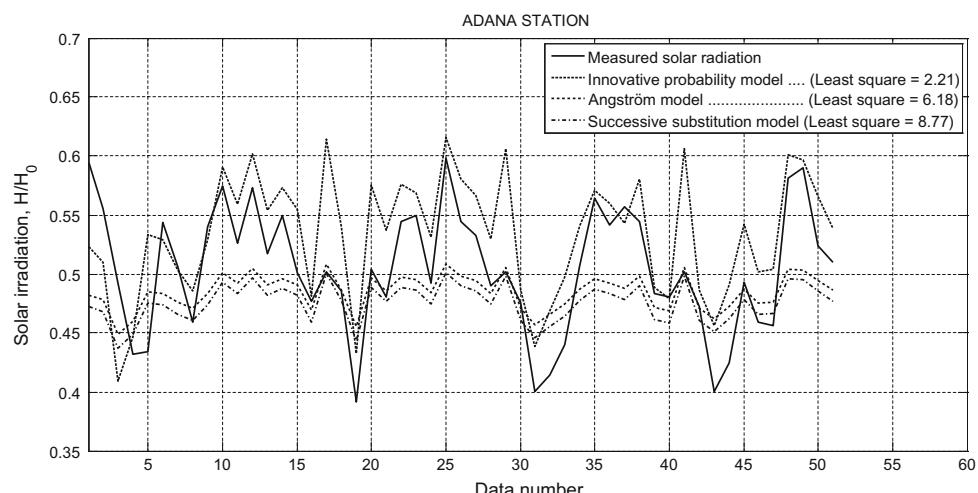
The wind speed measurement at a site is determined mainly by two factors—the overall weather systems (which usually have an extent of several hundred kilometers) and the nearby topography within few kilometers of the station. The application of measured wind speed statistics for calculating the wind energy potential in an area requires extrapolation of wind speed statistics to points at which measurements are not taken.

The collective effect of the terrain surface and obstacles, leading to an overall retarding of the wind near the ground, is referred to as the roughness of the terrain. Orographic elements such as hills, cliffs, ridges, and escarpments exert an additional influence on the wind. Roughness and orography are among the main factors that affect the wind speed.

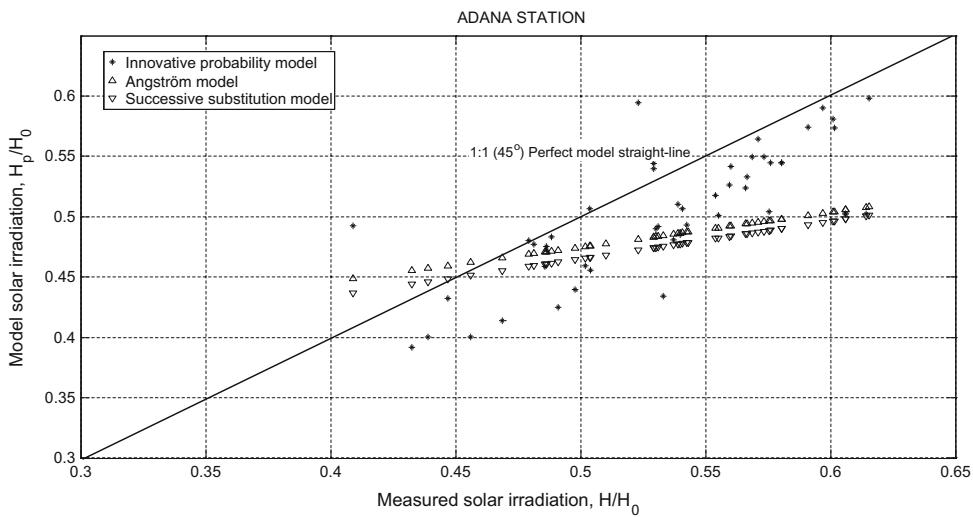
### 9.3.2 Power Formulations

Classical wind energy formulation is based on the kinetic energy definition whereby the mass is considered as a constant. Consequently, the wind energy is directly related to half (1/2) of the specific mass multiplication by the cube of

**Fig. 9.11** Performance of different methods



**Fig. 9.12** Model performance graph



wind velocity. The new approach by Şen (2013) is based first on the basic definition of force and then energy (work). The theoretical derivation of wind power initiates from the consideration of the kinetic energy definition in physics and for any given instantaneous wind speed,  $V$ , the instantaneous wind power,  $P$ , expression becomes as,

$$P = \frac{1}{2} \rho V^3 \quad (9.29)$$

where  $\rho$  is the density of air. Many researchers have assumed that the air density is independent of the wind speed cubed and constant as for the standard atmosphere being equal to  $1.293 \text{ kg/m}^3$  (Şen 1999).

Basically, wind energy is a version of kinetic energy because of the air movement during wind blows. Hence, it is possible to derive refined wind energy formulation by expressing the kinetic energy, and then subsequently the final formulation becomes (Şen 2013).

$$P = \frac{1}{3} \rho V^3 \quad (9.30)$$

It is obvious that the only difference between this expression and Eq. (9.29) as the conventional wind energy formulation is the numerical factor of (1/3) instead of (1/2). It is suggested that the use of Eq. (9.30) is more accurate, because the fluid property of the air is taken into consideration right from the beginning of the physical derivations.

### 9.3.3 Wind Field Data

In general, experts start to evaluate the long-term mean wind energy and the following steps are among the essentials for a successful accomplishment.

- (1) The raw data may not be reliable with many miss-recordings, recording errors, and others. Hence, the first step is to establish a reliable data base,
- (2) After the elimination of unreliable data parts, the raw data can be transferred to Microsoft Excel software,
- (3) The reliable data is adjusted for long-term wind fluctuations.

The raw data of 10-min measured averages are used for the present calculations after the reliability corrections. Table 9.3 includes the wind speed averages and standard deviations with coefficient of variation and the Weibull model parameters at field measurement height of 30 m with extrapolations to 70 m at a site (for İzmir-Çeşme-Ovacık meteorology station) in the western province of Turkey.

In any wind power calculation, it is necessary to know the theoretical wind speed PDF, which appeared as Weibull PDF with parameters given in this table. The following MATLAB program helps to generate the wind speed relative frequency.

**Table 9.3** Wind speed characteristics

Height (m)	Average speed (m/s)	Standard deviation (m/s)	Variation coefficient	Weibull PDF parameters	
				A	k
30	9.21	5.02	0.56	10.40	1.95
70	10.29	5.60	0.54	11.64	1.95

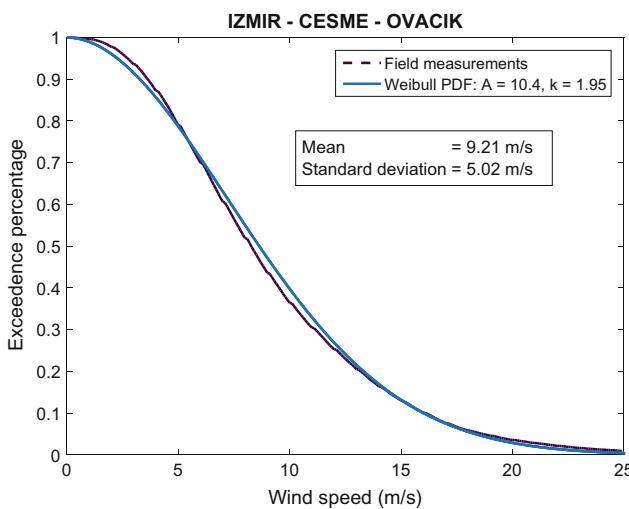
```

function wind (v,hcn,name)
% This program plots the time variation of the wind record at a site in addition to
% histogram and theoretical Weibull probability distribution function.
% v      = Wind speed
% hcn    = Histogram class number
nn=size(v);
n=nn(1,1);
figure (1)
plot(v);
xlabel ('Data sequence (time)')
ylabel ('Wind velocity (m/s)')
title (name)
% Statistical quantities
av=mean(v);
st=std(v);
vmax=max(v);
text(1,vmax-1,[' Mean      = ' num2str(av,3) ' m/s,' ' Std. dev. = ' num2str(st,3) ' m/s']);
% The following portion plots histogram
figure (2)
hist(v,hcn)
colormap white;
NH=hist(v,hcn);
xlabel('Wind velocity (m/s)')
ylabel('Frequency')
title(name)
a = wblfit(v(v>0));
x=0:0.1:vmax;
y=wblpdf(x,a(1,1),a(1,2));
hold on
plot(x,max(NH)/max(y)*y,'k','LineWidth',2)
text(0,max(NH),[' Weibull PDF   P_{1} = ' num2str(a(1),3) ',   P_{2} = ' num2str(a(2),3)]);
saveas(1,['Wind velocity ' name]);
saveas(2,['Histogram ' name]);
end

```

The execution of the above MATLAB program to İzmir, Ovacık 10-min wind data records yields field empirical and theoretical PDFs in Fig. 9.13.

One can observe that the field and theoretical Weibull PDF curves follow each other within practically acceptable errors.



**Fig. 9.13** Wind speed relative frequency and theoretical PDF

### 9.3.4 Turbine Calculations

In order to find the most suitable turbine for the wind velocity, one can try different industry productions. Herein as an example the NordexS70/1500 type of turbine is considered for use at Izmir Çesme, Ovacık location. The characteristics of such a turbine are given in the following with power curve values (Table 9.4).

**Nominal Power:** 1500 kW  
**Rotor diameter:** 70 m  
**Cut-in speed:** 3 m/s  
**Cut-out speed:** 20 m/s

The first step in any turbine calculation is to obtain the wind speed PDF curve, which is Weibull PDF at this site as already given in Fig. 9.13. The necessary MATLAB program for turbine yield is given below.

**Table 9.4** Power curve values

Speed (m/s)	Power (kW)	Speed (m/s)	Power (kW)
1	0	14	1500
2	0	15	1500
3	0	16	1500
4	25	17	1500
5	87	18	1500
6	214	19	1500
7	377	20	1500
8	589	21	0
9	855	22	0
10	1162	23	0
11	1453	24	0
12	1500	25	0
13	1500		

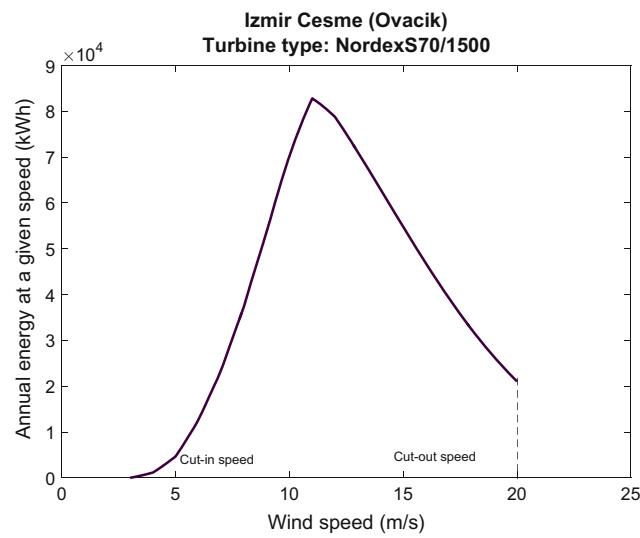
```

function TurbinGeneral(Type,cutin,cutoff,Scale, Shape)
% This program calculates the annual yield of the given turbine
% Type indicates the type of turbine
% Scale is the first Weibull probability distribution parameter
% Shape is the second Weibull probability distribution parameter
% v is the velocity that varies between 0 and 25 m/s
% freq is the annual wind blowing number according to Weibull probability
% distribution function
v1=0:0.1:cutin;
s=size(v1);
m=s(2);
v=0:0.1:cutoff;
s=size(v);
n=s(2);
freq(1)=wblcdf(v(1),Scale,Shape)*8760;
for i=m:n
freq(i)=8760*(wblcdf(v(i),Scale,Shape)-wblcdf(v(i-1),Scale,Shape));
end
Sturbin(m:n)=0;
for i=m:n
T=interp1(Type(:,1),Type(:,2),v(i));
Ti=T*freq(i);
Sturbin(i)=Sturbin(i)+Ti;
end
Sum(m)=Sturbin(m);
m1=m+1;
for i=m1:n
Sum(i)= Sum(i-1)+Sturbin(i);
end
plot(v(m:n),Sturbin(m:n))
figure(2)
plot(v(m:n),Sum(m:n))
Sum(n)
end

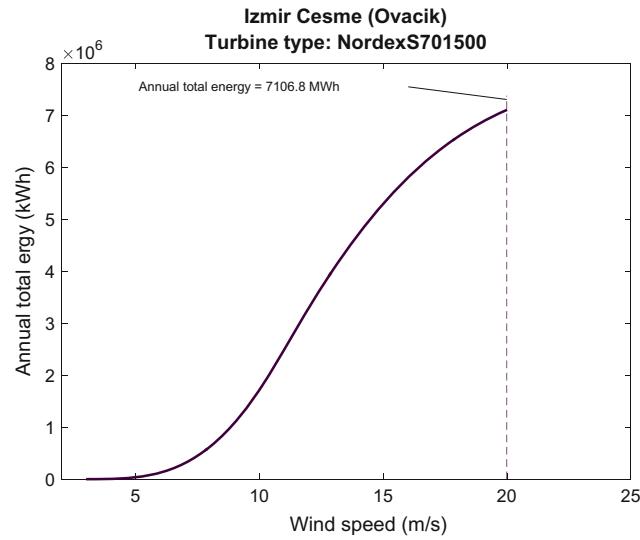
```

The wind energy calculations are made according to power curve values in Table 9.4 by considering aerodynamic principles. The resultant wind velocity variation according to the turbine type taken into consideration is

presented in Fig. 9.14. This graph provides the minimum (cut-in) and the maximum (cut-out) wind speed levels for the turbine considered in wing generation design.



**Fig. 9.14** Wind energy for a given velocity value



**Fig. 9.15** Annual total wind energy

The integration of the curve in Fig. 9.14 by numerical analysis yields the annual total wind energy as given in Fig. 9.15. The following MATLAB program provides the calculation of annual wind power.

A wind farm with 12 NordexS70/1500 turbines can generate at Ovacik location annually the following amount of energy.

$$12 \times 5756.5 = 69077.9 \text{ MWh/year}$$

```

function Turbin (Type,Scale, Shape)
% This program calculates the annual yield of the given turbine type
% Scale is the first Weibull probability distribution parameter
% Shape is the second Weibull probability distribution parameter
% v is the velocity that varies between 0 and 25 m/s
% freq is the annual wind blowing number according to Weibull probability
% distribution function
v=0:0.1:25;
freq(1)=wblcdf(v(1),Scale,Shape)*8760;
for i=2:251
    freq(i)=8760*(wblcdf(v(i),Scale,Shape)-wblcdf(v(i-1),Scale,Shape));
end
Sturbin(1:251)=0;
Sum(1)=0;
for i=2:251
    T=interp1(Type(:,1),Type(:,2),v(i));
    Ti=T*freq(i);
    Sturbin(i)=Sturbin(i)+Ti;
    Sum(i)=Sum(i-1)+Ti;
end
plot(v(1:251),Sturbin(1:251))
figure(2)
plot(v(1:251),Sum(1:251))
Sum(251)
end

```

**Table 9.5** Single turbine energy generation characteristics

WEC type	Power (kW)	Rotor (m)	Hub height (m)	Single turbine (MWh)	Parking efficiency (%)	Single turbine in park (MWh)
NordexS70/1500	1500	70	65	7106.8	90	6396.1

The single turbine of NordexS70/1500 type yields the following values as in Table 9.5.

Based on the consideration of 10% total uncertainty, the wind energy output of a turbine in a wind farm becomes

$$6396.1 \times 0.9 = 5756.5 \text{ MWh}$$

At this location the total wind energy for generation is 18 MW, and therefore, the number of NordexS70/1500 turbines is  $18/1.5 = 12$ .

#### 9.3.4.1 Power

In any wind study for application one should care for the quantitative amounts of practical and theoretical wind power variation within the wind speed data range. In Fig. 9.16, these variations are presented and it is obvious that the practical turbine power output is always less than the theoretical calculations.

Another version of the previous figure is in the form of cumulative power curves, which is more reflective of power

generation according to wind speed (Fig. 9.17). The following MATLAB program provides the calculation of total annual yield.

Especially at medium wind speeds, the difference between the two becomes greater. The necessary MATLAB program for the calculations is given below.

```

function windcross (v,name)
% This program calculates the wind velocity continuity curve
% v = Wind velocity
cr(1:251)=0:0.1:25;
cn(1:250)=0;
for i=1:250
c=v(v>cr(i));
cn(i)=cn(i)+length(c);
end
cm=max(cn);
cmp=cn/cm;
av=mean(v);
st=std(v);
figure (1)
plot(cr(1:250),cmp);
xlabel('Wind velocity (m/s)')
ylabel('Wind velocity exceedence percentage')
title('Wind continuity curve')
text(10,0.9,['Ort. =' num2str(av,3), ' m/s' , St. sap. = ' num2str(st,3), ' m/s']);
hold on
a=wblfit(v(v>0));
mv=max(v);
x=0:0.01:mv;
y=wblcdf(x,a(1,1),a(1,2));
plot(x,1-y,'k')
text(10,max(cmp)-0.2,[' Weibull PDF P_{1} = ' num2str(a(1),3) ' , P_{2} = ' num2str(a(2),3)]);
sum(1:250)=0;
for i=2:250
p(i)=0.5*1.23*cmp(i)*cr(i)^3;
pt(i)=0.59*p(i);
sum(i)=sum(i-1)+pt(i);
end
maxsum=max(sum);
figure (2)
plot(cr(1:250),p)
xlabel('Wind velocity (m/s)')
% Theoretical wind power calculations
ylabel('Theoretical wind power (W/m^2)')
title('Theoretical wind power')
hold on
plot(cr(1:250),pt)
xlabel('Wind velocity (m/s)')
% Practical wind power calculations
ylabel('Practical wind velocity (W/m^2)')
title('Practical wind velocity')
figure (3)
plot(cr(1:250),sum)
xlabel('Wind velocity (m/s)')
ylabel('Total practical wind power (W/m^2)')
title('Total practical wind power')
text(1,maxsum-50,[' Total wind power ' num2str(maxsum,3), ' Watt/m^2']);
end

```

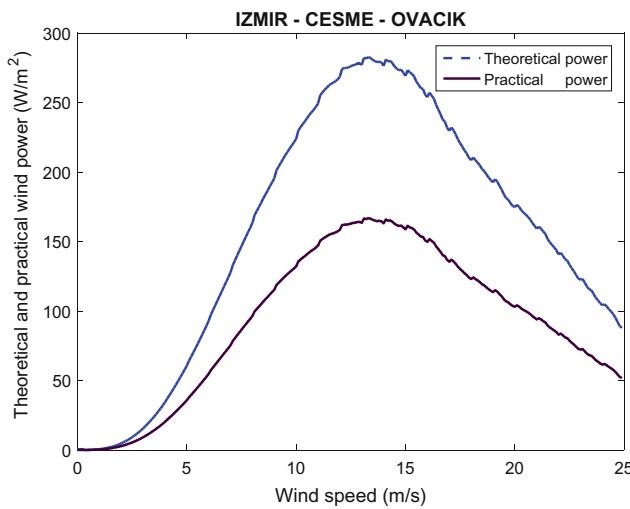


Fig. 9.16 Practical wind power

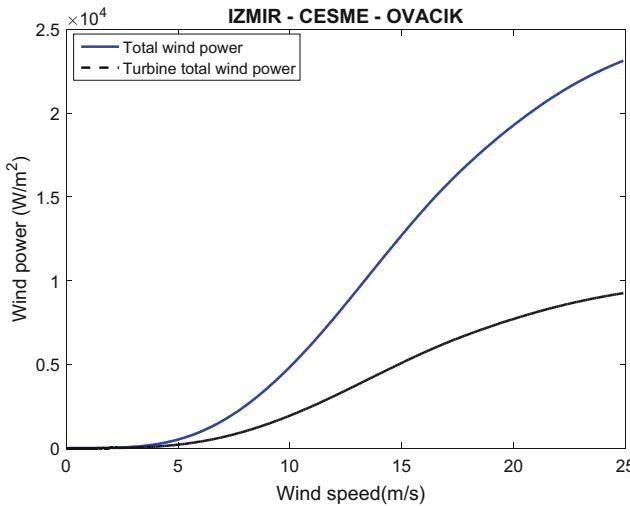


Fig. 9.17 Total wind power

## 9.4 Hydropower

Hydropower generation is significant for energy supply especially in the climate convenient regions of the world, where there are frequent storm rainfalls that provide support for surface flow in the rivers. Dam construction across a convenient section of the river provides water storage with potential energy head, which is then converted to kinetic energy by the water turbines first to hydroelectric power (HP) production and then through generators into electrical energy. The fundamentals and systems of hydropower are presented based on social, engineering, climatic, hydrological, and technological points. Natural hydrological cycle is the sole source of hydropower (HP) as a renewable energy

alternative in different parts of the world. HP plant designs have significant flexibility for peak demand supply with high and low capacity factors. For the time being, HP is the largest renewable energy source with about 20 % share in the world's electricity and over three-fifths of the world's renewable electricity. The hydroelectricity extraction from water depends not only on the water volume (discharge, which is the volume of water per time duration), but also on the difference in height between the water head elevations (the height difference) and the water outflow, which is generally the elevation of the turbines. The amount of water potential energy is directly proportional to the head and also the discharge of water that enters the turbines by either a penstock or pipes.

### 9.4.1 Classical and Energy Tree Hydropower Models

The power production,  $P$ , can be calculated by taking into account various impactive factors as (Alashan et al. 2016),

$$P = \eta \gamma g Q H \quad (9.31)$$

Here,  $\gamma$  is the density of water;  $H$  is the water head,  $Q$  is the flow rate (discharge);  $g$  is the acceleration of gravity; and  $\eta$  is an efficiency coefficient that varies between 0 and 1. Efficiency is the highest for installations in case of large and modern turbines.

There are two classical, namely single- and double-point and innovative energy tree, alternatives for hydropower calculation methodologies.

The single point is the most commonly used practical formulation for preliminary HP, as  $\gamma$  the specific water weight in kN/m<sup>3</sup>,  $Q$  the discharge in m<sup>3</sup>/s, and  $H$  the falling head in m.

$$P = \gamma Q H \quad (9.32)$$

The end product,  $P$ , is in kW. In order to assess the gross hydroelectric power potential of a drainage basin, most often the height and the discharge are taken at the outlet point of drainage basin concerned.

As for the double-point methodology, Eq. (9.32) does not take into consideration any geomorphological parameters of the drainage basin such as the elevation differences along the main channel, and therefore, the results cannot be reliable. Height and discharge measurement considerations at two sites provide better results. In practice, two points are taken as the arithmetic average of the drainage basin height,  $h_a$ , at the centroid point and another height,  $h_o$ , at the outlet point. Hence, rational and logical reasoning leads to another version of Eq. (9.32) as (Alashan et al. 2016),

$$P = \gamma(h_a - h_o) \frac{Q_a + Q_o}{2} \quad (9.33)$$

Herein,  $Q_a$  and  $Q_o$  are the discharges at the centroid and outlet points, respectively. This method has the following implicit assumptions.

- (1) The elevation difference assumes a linear height change between the two points, whereas most often in nature there are nonlinear topographic variations between the two points,
- (2) The arithmetic average of the two discharges implies that there is a constant discharge, whereas in nature depending on the geomorphological features of the drainage basin between the centroid and outlet locations, the discharge accumulates with distance along the main channel and reaches its maximum value at the outlet,
- (3) Although the height difference is taken into consideration, this formulation does not consider the distance between the points.

If for a drainage basin or any part of it, the water entrance (input) and exit (output) points' elevations ( $h_i$  and  $h_o$ ) are considered with input and output discharges ( $Q_i$  and  $Q_o$ ), then the gross HP can be thought of two parts.

- (1) The input discharge generates hydroelectric power along the main channel until it reaches the output point, and during its travel the discharge falls from the height equal to  $(h_i - h_o)$ ,
- (2) The discharge amount that comes from the surface area of the drainage basin is equal to  $(Q_o - Q_i)$ . As for the falling head of this discharge the difference between the average drainage basin height,  $h_a$ , and the outlet height,  $h_o$ , is considered as  $(h_a - h_o)$  in Eq. (9.33). Hence, another classical alternative for hydropower calculation emerges as follows.

$$P = \gamma(h_a - h_o)(Q_0 - Q_i) + \gamma(h_i - h_o)Q_i \quad (9.34)$$

The weighted average elevation value can be calculated according to the following expression, which is based on the hypsographical curve (HC) concept.

$$h_a = \frac{\sum_{i=1}^n A_i h_i}{\sum_{i=1}^n A_i} \quad (9.35)$$

or for continuous data availability case according to,

$$h_a = \frac{\int_{A_i}^{A_o} h(A) dA}{A_o - A_i} \quad (9.36)$$

where  $A_i$  and  $h_i$  are the area and elevation of  $i$ -th drainage sub-area;  $n$  is the number of sub-drainage areas. This method yields better results in drainage basins with low flow potentials.

The energy tree (ET) is a new, innovative, and more effective approach suggested by Alashan et al. (2016). The basic equation of ET includes two important parameters, the flow discharge and elevation difference, which are simply related to the summation of HP formulation as,

$$P = \gamma \sum_{i=1}^N Q_i (H_i - H_{i-1}) \quad (9.37)$$

Herein,  $Q_i$  is the mean flow at  $i$ th location with the elevation,  $h_i$ , at the downstream location and  $h_{i-1}$  is the previous upstream point. Theoretically, Eq. (9.37) can be converted into an integration form as,

$$P = \gamma H \int_{H_d}^{H_u} Q(h) dh \quad (9.38)$$

This expression reflects the most refined hydroelectric power calculation, because the discharge along the main channel is dependent on the elevations and also discharges from the sub-basins in a continuous manner. In Eq. (9.38), the discharge is shown as the function of elevation. For the application of ET methodology, the following steps are necessary.

## 9.5 Newtonian Energy Principle

The change in force,  $dF$ , is defined as the change of momentum in the Newtonian physics as,

$$F = \frac{d(mv)}{dt} \quad (9.39)$$

where  $mv$  is the momentum,  $m$  is the mass, and  $v$  is the velocity of this mass. It is well known that absorbed radiation of energy is accompanied (in classical theory, in quantum theory, and in experiment) by a radiation momentum, which is the ratio of energy,  $E$ , to light velocity,  $c$ , as  $E/c = mc$  (Shadowitz, 1968). Planck's formula for energy is  $E = h$ , where  $h$  is Planck's coefficient,  $v$  is frequency and it describes energy equivalence on a quantum basis (Planck 1900). Another energy equivalence formula is given by Einstein (1905a, b). In the derivation of Einstein's equation, conservation of momentum is used, and likewise, herein, the change of momentum is taken as a basic approach to this problem simply in the Newtonian physics domain. Energy transfer can be defined as work. The change

in the kinetic energy of a mass is equal to the net work done on the mass. Energy is defined also as the capacity for doing work. Hence, energy change,  $dE$ , is equivalent to work change,  $dE = FdL$ , where  $dL$  is the distance covered for movement. After multiplying both sides of Eq. (9.39) by  $dL$  and considering the velocity definition as,  $v = dL/dt$ , leads to,

$$dE = d(mv)v \quad (9.40)$$

The right-hand side of this expression can be expanded as,

$$dE = v^2 dm + mv dv \quad (9.41)$$

this indicates that energy variation consists of two gradients.

- (1) Change in mass (the first term on the right-hand side is similar to Einstein's energy,  $dE = c^2 dm$ ),
- (2) Change in velocity (the second term on the right-hand side, related to Newtonian kinetic energy change).

Classical integration of Eq. (9.41) leads to the following expression, which has not been seen in the literature previously.

$$E = v^2 m + \frac{1}{2} m v^2 = \frac{3}{2} m v^2 \quad (9.42)$$

The integration is taken from zero to a constant upper limit for each variable as  $E$ ,  $m$ , and  $v$ . For light velocity, it yields,

$$E = \frac{3}{2} m c^2$$

this is different than Einstein's energy equation. However, philosophically and logically there is conceptual mistake in this expression, and therefore, although it is mathematically correct, it is physically not valid due to the fact that continuous and simultaneous increase of velocity and mass is not realistic starting from zero.

Mass and energy are regarded as distinct properties, because in Newtonian physics and as in Eq. (9.41) they are distinctively measured in different units. This is due to the fact that spatial and temporal units are perceived separately, which gives rise to the perception of different mass and time properties. Equation (9.41) can be expressed verbally as,

Energy change is equal to the summation of a constant (velocity square,  $v^2$ , assumed constant) times change in mass plus another constant (mass,  $m$ , assumed constant) times velocity and velocity change

This expression includes both large-scale (relativity) and ordinary-scale (common sense) perceptions. To this end, the following two questions can be asked.

- (1) What is the energy,  $E_c$ , if the velocity (light velocity) is constant? This case corresponds to the assumption of special relativity where the velocity is assumed constant as light velocity,  $c$ , only (Einstein 1905a, b). The answer is that the second term in Eq. (9.41) becomes zero because there is no velocity change, and hence, the energy remains as,

$$dE_c = c^2 dm$$

This expression gives the total energy,  $E_c$ , due to mass change after integration as,

$$E_u = mc^2 \quad (9.43)$$

This is the most well-known Einstein's energy equation in the special theory of relativity. According to special relativity, light travels at the same speed for all inertial observers, and this implies that one can select units such that spatial distances are specified by units of time (space-time concept). In such units energy and mass have the same units and they are equal numerically, which implies that mass and energy are not two distinct properties. In a way, the perception of mass and energy as distinct units is due to the fact that spatio-temporal intervals are overlooked.

- (2) The second question is what is the energy,  $E_m$ , if the mass is constant? In this case, the first term on the right-hand side of Eq. (9.41) is zero and the energy expression takes the form as,

$$dE_m = mv dv \quad (9.44)$$

which after the integration yields the total energy due to the velocity change only as,

$$E_m = \frac{1}{2} m v^2 \quad (9.45)$$

This is the kinetic energy expression in the Newtonian physics domain.

After all these simple derivations, Eq. (9.41) obviously indicates that mass and energy are not the same as suggested by some philosophers and physicists. The same equation implies that mass and energy are distinct properties of physical systems.

## References

- Alashan, S., Şen, Z., and Toprak, Z.F., (2016). Hydroelectric energy potential of Turkey: A refined calculation method. *Arabian J. Sci. Engg.*, Vol. 41(4): 1511–1520.
- Akinoglu, B.G., Ecevit, A., (1990). Construction of a quadratic model using modified Angström coefficients to estimate global solar radiation. *Sol Energy*, Vol. 45:85–92.
- Angstrom, A., (1924). Solar and atmospheric radiation. *Q J R Meteor Soc* 50:121.
- Benjamin, J.R., and Cornell, C.A., (1970). Probability, Statistics and Decision Making in Civil Engineering. McGraw-Hill: 684 p.
- Caughey, S.J., Crease, B.A., and Roach, W.T., (1982). A field study of nocturnal stratocumulus II. turbulence structure and environment. *Quart J R Meteor Soc.*, Vol. 108:125–44.
- Davies, R., (1978). The effect of finite geometry on three dimensional transfer of solar irradiance in clouds. *J Atmos Sci.*, Vol. 35:1712–25.
- Einstein, A. (1905a) Does the inertia of a body depend upon its energy content? In: A. Einstein, H. Minkowski and H. Weyl (Eds.) *The principles of relativity*, pp. 69–71 (1952). New York.
- Einstein, A., (1905b). On the electrodynamics of moving bodies. *Ann. Phys.* 17:891.
- Graneau, P., (2005). The Failure of  $E = mc^2$ . The 2005 John E. Chappell Memorial Lecture of the Natural Philosophy Alliance, Issue 61, Infinite Energy. presented at the NPA 12th Annual Conference, May 2005.
- Gumbel, E.J., (1958). Statistics of Extremes. Colombia University Press, New York.
- Joseph, J.H., and Cahalan, R.F.. (1988). Nearest neighbor spacing in four weather cumulus as inferred from Landsat. *J Clim Appl Meteor*.
- Planck, M., (1900). Zur Theorie des Gesetzes der Energieverteilung im Normalspektrum. *Verhandl. Deutsch. Phys. Ges.*, Vol. 2: 237.
- Şahin, A.D., and Şen, Z., (1998). Statistical analysis of the Angström formula coefficients and application for Turkey. *Solar Energy*, Vol. 69: 29–38.
- Şen, Z., (1999). Terrain Topography Classification for Wind Energy Generation. *Renewable Energy*, **16**, 904–907.
- Şen, Z., (2001a). Angström equation parameter estimation by unrestricted method. *Solar Energy*, Vol. 71:95–107.
- Şen, Z., (2001b). Areal assessment of wind speed and topography with applications in Turkey. *Renewable Energy*, Vol. 24: 113–129.
- Şen, Z., (2004). Solar energy in progress and future research trends. *Progress in Energy and Combustion Science*, Vol. 30: 367–416.
- Şen, Z., (2007). Simple nonlinear solar irradiation estimation model. *Renewable Energy*, Vol. 32: 342–350.
- Şen, Z., (2008). *Solar Energy Fundamentals and Modeling Techniques. Atmosphere, Environment, Climate Changer and Renewable Energy*. Springer Verlag, 276 pp.
- Şen, Z., (2013). Innovative Wind Energy Models and Prediction Methodologies. In *Energy Systems Handbook of Wind Power Systems*, Springer, 708 pp.
- Şen, Z., (2016). Probabilistic innovative solar irradiation estimation. *Int. J. Energy Res.*, Vol. 41(2): 1–11.
- Shdowitz, A. (1968) *Special Relativity*, Dover Publications, Inc. New York, pp. 203.



## 10.1 General

Any time or space series may have structurally a linear or nonlinear deterministic component with a mathematical form throughout the whole record period. These components are the best indicators for general tendency embedded in the series records, which show increasing or decreasing inclinations that are very significant for internal or external (future, projection) estimations. It is well known in the literature that a given time series may have periodicity, jump (abrupt) change, trend, and stochastic component (Cox and Miller 1965; Box and Jenkins 1970). In monthly earth systems records, periodicity takes place due to astronomical events, but trend component is frequently sought component, because it is mostly related to human activities (anthropogenic effects).

Trend analysis evaluation is also needed for long-term infrastructure design and risk analysis in hydro-meteorological time series. Especially, with the concept of climate change assessment, trend identification, detection, and evaluation are important issues in different disciplines. The growing importance of trend analysis is triggered also by the Intergovernmental Panel on Climate Change reports (IPCC 2007, 2013).

The literature is full of many applications in different parts of the world, but they all provide overwhelmingly holistic trends within the given time series. In general, trend identification is achieved either by a moving average procedure or by classical calculation methodologies. Mann-Kendall (Mann 1945; Kendall 1975) trend test analysis is applied for trend start and existence identification with trend slope calculation Sen (1968). These methodologies search for holistic and absolute trends without any distinction between “low,” “medium,” and “high” clusters in the trend search.

Nonparametric Mann-Kendall (Mann 1945; Kendall 1975, MK) statistical test is used since the last several decades in search for trends in the past records in order to be able to understand the environmental conditions such as

water pollution, climate change, and global warming. This test assumes that the records have independent serial structure, since dependence (positive serial correlation structure) may lead to trend detections in the absence of trends. Although a pre-whitening procedure application is proposed prior to MK test application for rendering the original series into independent series (von Storch 1995), it has been shown by Douglas et al. (2000) that such a pre-whitening may lead to trend detection that is less than the one in the original series. A detailed account of pre-whitening procedure prior to trend detection works has been presented by Yue and Wang (2004). They concluded that pre-whitening is not suitable for estimating the effects of serial correlation on the MK test when trend exists within a time series. However, recently, over-whitening procedure is shown to be more effective than the pre-whitening in search of possible trends in a time series (Sen 2016).

This chapter suggests partial trend analysis in the sense that trends are sought with respect to some baseline and few clusters within time series. The main purpose of this section is to introduce rather partial trend analysis by two, three, or multiple classes. Few new trend analysis methodologies are also presented. Furthermore, this chapter presents the applicability of a partial trend analysis based on the ordered sub-samples from the same time series without any pre-treatment such as the pre-whitening procedure. Last but not least, several MATLAB programs are presented that are very helpful in automatic trend identifications.

## 10.2 Trend Concepts

All the classical trend tests have the following basic ingredients in their applications, and their existence does not provide a fruitful floor for significant trend identification.

- (1) The classical trend methodologies search for monotonic trend component in a holistic manner without any distinction among “low,” “medium,” and “high” values,

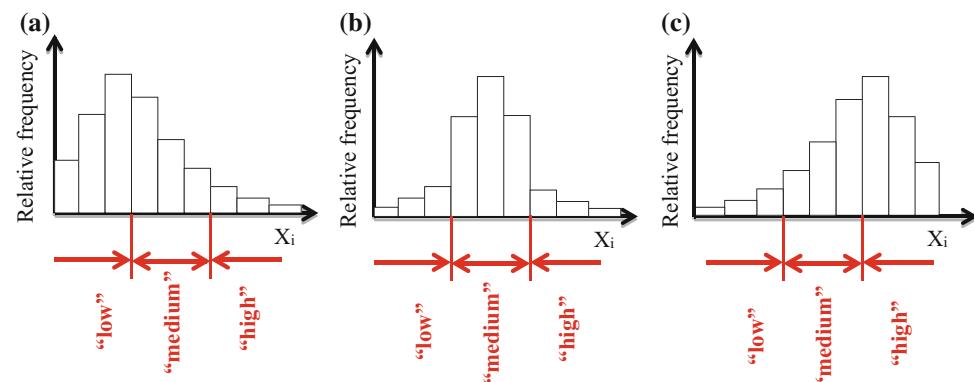
- (2) The search is in the real-time record period of the series, and hence, the serial correlation coefficient becomes important,
- (3) The correlation structure of any given series is assumed independent or at least has closely independent form,
- (4) It is possible to apply pre-whitening procedure to render the serially dependent time series structure into an independent form, but the pre-whitening procedure disturbs the original structure of the time series,
- (5) The results are affected by finite length time series.

The study by Yue et al. (2002) evaluated the MK and Spearman rho tests power through Monte Carlo simulations, and they stated that the power of these tests depends on the pre-assigned significance level, trend magnitude, sample size, and the amount of variation within a time series. Hence, the bigger the absolute magnitude of trend, the more powerful are these tests; as the sample size increases the tests become more powerful; and as the amount of variation increases within a time series, the power of the tests decreases. Fatichi et al (2009) examined trends in time series of average temperature records by using different trend test methods. They concluded that an increase occurs in uncertainty when pronounced stochastic behaviors are present in the data.

A given time series,  $X_i$  ( $i = 1, 2, \dots, n$ ) has overall probability distribution function (PDF) in the forms of symmetrical, left (negatively), and right (positively) biased histograms (Fig. 10.1).

The relative frequency is equivalent with the probability definition. It shows the change of time series classification percentages in a set of classes. Each class is a categorization. It is possible to thing the number of categorizations as two-class, verbally, as “low” and “high,” or as three-class “low,” “medium,” and “high,” which are shown in Fig. 10.1 or any convenient number of classes. Especially, in three verbal divisions “low” values are representatives of droughts, “medium” class are for “management,” and “high” class implies extreme events such as “flood” occurrences.

**Fig. 10.1** Histograms:  
a positively biased,  
b symmetrical, c negatively  
biased



The most important question at this stage is whether to search for a holistic monotonic trend or better to look for trend in each category. The answer is, of course, triple trend is better than monotonic search. The literature is full of monotonic trend analyses by means of the classical trend analysis methodologies.

A good example for two-class trend identification is the most frequently used climate change fundamental definition in terms of theoretical PDFs as in Fig. 10.2.

After the comparison of two PDFs with shifts in the arithmetic average and standard deviation, this figure indicates obviously that there are two classes. In the climate change explanations, these shifts are presented implicitly as trend component. The slope of possible trend from these two graphs can be calculated as the differences divided by the half number of time series length (Şen 2015a, b). The trend concerned with the arithmetic average is given as,

$$S_\mu = \frac{\mu_s - \mu_f}{n/2} \quad (10.1)$$

Similarly for the standard deviation trend, which is referred to as the variability (Şen 2017), the slope is calculated as,

$$S_\sigma = \frac{\sigma_s - \sigma_f}{n/2} \quad (10.2)$$

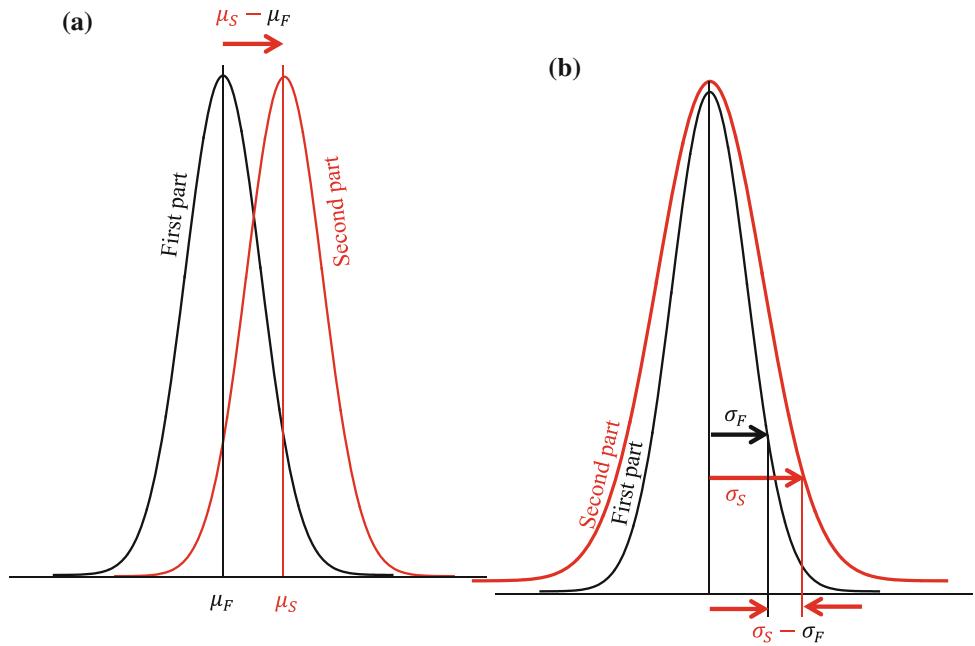
In case of multiple class, Fig. 10.2 can be generalized with a set of arithmetic average and standard deviation trend slopes between two successive classes. Hence, the generalization of Eqs. (10.1) and (10.2) can be written as,

$$S_{\mu i} = \frac{\mu_{si} - \mu_{fi}}{n_i} \quad (10.3)$$

and

$$S_{\sigma i} = \frac{\sigma_{si} - \sigma_{fi}}{n_i} \quad (10.4)$$

**Fig. 10.2** Monotonic trends:  
**a** trend in arithmetic average,  
**b** trend in standard deviation



respectively, where  $n_i$  is the number of terms in one of the sub-time series. For instance, in triple trend analysis,  $i = 3$ , and hence,  $n_i = n/3$ .

### 10.3 Innovative Trend Template (ITT) Analysis

Efficient, effective, and optimum management of water resources requires the identification of trends not only holistically over a given time period, but also whether the “low,” “medium,” and “high” values have separate trends. These help to identify drought and flood occurrences in their decreasing or increasing frequencies. In case of a holistic trend, there is a gradual change over the whole record period and it is expected to continue in the future. However, for “low,” “medium,” or “high” value trend search, the periods are comparatively shorter. As Zhang et al. (2001) suggest the hydrological literature has so far devoted very limited attention to the characterization of trend pattern. In general, abrupt or gradual trend patterns are sought in a given time series. The ITT methodology helps to identify possible

trends in “low,” “medium,” and “high” time series values separately and objectively. It is a square area, which is divided into two triangular halves by the 1:1 ( $45^\circ$ ) straight-line as in Fig. 10.3 (Sen 2012, 2014). The horizontal axis is for the first half ordered time series, and the vertical axis represents the second half.

This simple trend identification methodology is straightforward to apply, and depending on the scatter of points, one is able to make a series of interpretations about any set of groups such as “low” and “high” time series values for extreme events (droughts, floods); “moderate” values for better management studies. Apart from the scatter of points, in order to understand the general trend slope whether increasing or decreasing, one can plot the arithmetic averages of the first and second halves on the same graph and, accordingly, make decision about the trend type. In case of the arithmetic averages of the two halves may fall on the 1:1 ( $45^\circ$ ) line or very close to it, then the final decision is that there is no trend in the given time series. The slope of trend can be calculated from Eq. (10.1) quantitatively.

In the following trend simulation MATLAB program is presented with possibility of selection various PDFs.

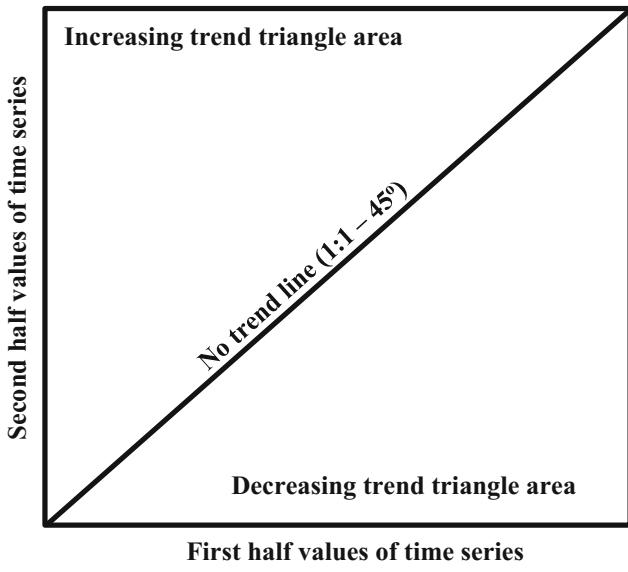
```

function [XT] = TrendSimulation(A,N,Mean,StDev)
% THIS PROGRAM IS WRITTEN BY ZEKÂİ ŞEN ON 3 MARCH 2012
% IT SIMULATES TIME SERIES OF N SAMPLE LENGTH WITH 19 DIFFERENT CORRELATION
% COEFFICIENTS, 19 DIFFERENT TRENS SLOPES WITH MEAN, Mean AND STANDARD
% DEVIATION, StDev AS FOLLOWS:
% N = Length of simulation time series
% Rho = The correlation coefficient
%
% A = name of the distribution function as follows
%   'beta' or 'Beta',
%   'bino' or 'Binomial',
%   'chi2' or 'Chisquare',
%   'exp' or 'Exponential',
%   'ev' or 'Extreme Value',
%   'f' or 'F',
%   'gam' or 'Gamma',
%   'gev' or 'Generalized Extreme Value',
%   'gp' or 'Generalized Pareto',
%   'geo' or 'Geometric',
%   'hyge' or 'Hypergeometric',
%   'logn' or 'Lognormal',
%   'nbin' or 'Negative Binomial',
%   'ncf' or 'Noncentral F',
%   'nct' or 'Noncentral t',
%   'ncx2' or 'Noncentral Chi-square',
%   'norm' or 'Normal',
%   'poiss' or 'Poisson',
%   'rayl' or 'Rayleigh',
%   't' or 'T',
%   'unif' or 'Uniform',
%   'unid' or 'Discrete Uniform',
%   'wbl' or 'Weibull'
%
% Slope = trend slopes
%
% EXPLANATIONS
% NO  Rho    Slope
1 : -0.9  -0.009
2 : -0.8  -0.008
3 : -0.7  -0.007
4 : -0.6  -0.006
5 : -0.5  -0.005
6 : -0.4  -0.004
7 : -0.3  -0.003
8 : -0.2  -0.002
9 : -0.1  -0.001
10 : 0.0  -0.000 Independence and no trend
11 : 0.1  0.001
12 : 0.2  0.002
13 : 0.3  0.003
14 : 0.4  0.004
15 : 0.5  0.005
16 : 0.6  0.006
17 : 0.7  0.007
18 : 0.8  0.008
19 : 0.9  0.009

```

..... Continued .....

```
% MAIN PROGRAM
%
Rho=-1.0;
for i=1:19 % autocorrelation coefficient loop
    r=random(A,Mean,StDev,[N,1]);
    Rho=Rho+0.1;
    Rho2=Rho*Rho;
    X(1)=Mean+StDev*r(1);
    for j=2:N % Lag-one Markov process generation
        X(j)=Mean+Rho*(X(j-1)-Mean)+StDev*sqrt(1-Rho2)*r(j);
    end
    Slope=-0.01;
    for k=1:19 % slope loop
        Slope=Slope+0.001;
        for l=1:N % Trend incorporation
            XT(i,k,l)=X(l)+Slope*l; % i for correlation coefficient; j for trend
            % slope; and l for incorporation
        end
    end
end
for i=1:1:19
    figure
    xlabel('First half of time series')
    ylabel('Second half of time series')
    title(['Correton coefficien = '])
    for j=1:1:19
        scatter(sort(XT(i,j,1:N/2)),sort(XT(i,j,(N/2+1):N)), 'k.')
        hold on
    end
end
end
```



**Fig. 10.3** Innovative trend templates

## 10.4 Innovative Variability Analysis

As explained earlier in Sect. 10.2 through Fig. 10.2b, the meaning of variability as used in this section is whether there is any trend in the standard deviation along a given time series. The same ITT is also used for deducing the possible trend tendency in the standard deviation of the given time series after plotting the standard deviation values of the first and second parts. Similar to the arithmetic average case, the closer is the standard deviation point to the no-trend straight-line the variability trend slope is small, but in case of appreciably close point to 1:1 line or on it, there is no significant variability in the given time series. Equation (10.2) provides numerical calculation possibility for variability trend slope. The calculations can be achieved by the following MATLAB program with subsequent figures and their explanations.

```

function [MSlope,SSlope] = ZekaiSenInnovativeTrendAnalysis(X,Y,Title,XTitle,YTitle)
% This program provides 1:1 trend figure and all the necessary quantities
% for a given time series
% (written by Zekai Sen on 25 January 2012)
% X = Given time series
% Y = Time sequence for the time series
% m = average
% s = standard deviation
% N = number of data
% MSlope = Mean trend Slope
% SSlope = Standard deviation slope
n=floor(length(X));
n2=floor(n/2);
nm=mean(Y);
m=mean(X);
s=std(X);
FH=sort(X(1:n2,1));
SH=sort(X(n2+1:2*n2,1));
figure
scatter(FH,SH,'k*');
hold on
scatter(mean(FH),mean(SH),'rv');
scatter(m+std(FH),m+std(SH),'r^');
Xmin=min([min(FH),min(SH)]);
Xmax=max([max(FH),max(SH)]);
axis([Xmin Xmax Xmin Xmax]);
box on
grid on
% axis equal
title(Title)
 xlabel('1881 - 1947')
 ylabel('1948 - 2013')
 FHfm=mean(FH(FH<m));
 SHfm=mean(SH(SH<m));
 FHsm=mean(FH(FH>m));
 SHsm=mean(SH(SH>m));
 Slope=(SHsm-SHfm) / (FHsm-FHfm);
 % Inter=SHfm-Slope*FHfm;
 Inter=mean(X)-Slope*nm;
 scatter(FHfm,SHfm,'r<');
 scatter(FHsm,SHsm,'r>');
 legend('Location','SouthEast','Data points','Mean centroid','St. Dev. centroid','First half centroid','Second half centroid')
 line([Xmin Xmax],[Xmin Xmax],'Color','k','LineWidth',3);
% line([Xmin Xmax],[Inter+Slope*Xmin Inter+Slope*Xmax],'Color','k','LineWidth',1);
 MSlope=2*(mean(SH)-mean(FH))/n;
 SSlope=2*(std(SH)-std(FH))/n;
figure
plot(Y,X,'k')
grid on
title(Title)
 xlabel(XTitle)
 ylabel(YTitle)
hold on
scatter(nm,m,'r')
Mfin=m+MSlope*n2;
Minf=m-MSlope*n2;
May=max(Y);
Miy=min(Y);
line([Miy May],[Minf Mfin],'LineWidth',2,'Color','k','LineStyle','--')
scatter(nm,m+s,'r^')
Sufin=m+s+SSlope*n2;
Suini=m+s-SSlope*n2;
Max=max(X);
text(Miy,Max-5,['Trend slope      = ' num2str(MSlope),', Intercept =',num2str(Mini),'^oC'])
text(Miy,Max-15,['Variability slope = ' num2str(SSlope),', Intercept =',num2str(Suini),'^oC'])
line([Miy May],[Suini Sufin],'LineWidth',1,'Color','k','LineStyle','--')
legend('Location','SouthEast','Time series','Trend centroid','Trend line (Arithmetic average)','Variability centroid','Variability line (Standard deviation)')
end

```

The two most important points in the use of this software are the following statements that need user interference.

- (1) The duration of time series should be entered in two equal parts as indicated in the program,

```
a) xlabel('1881 - 1947')
b) ylabel('1948 - 2013')
```

In this example the start (end) of time series are 1881 and 2013, respectively.

- (2) Trend and variability intercepts are given in °C unit, but they must be changed according to the unit of entrance data. The statements in the software are.

```
a) text(Miy,Max-5,['Trend slope      = '
num2str(MSlope), '           Intercept
= ',num2str(Mini), '^oC'])
b) text(Miy,Max-15,['Variability slope = '
num2str(SSlope), '           Intercept
= ',num2str(Suini), '^oC'])
```

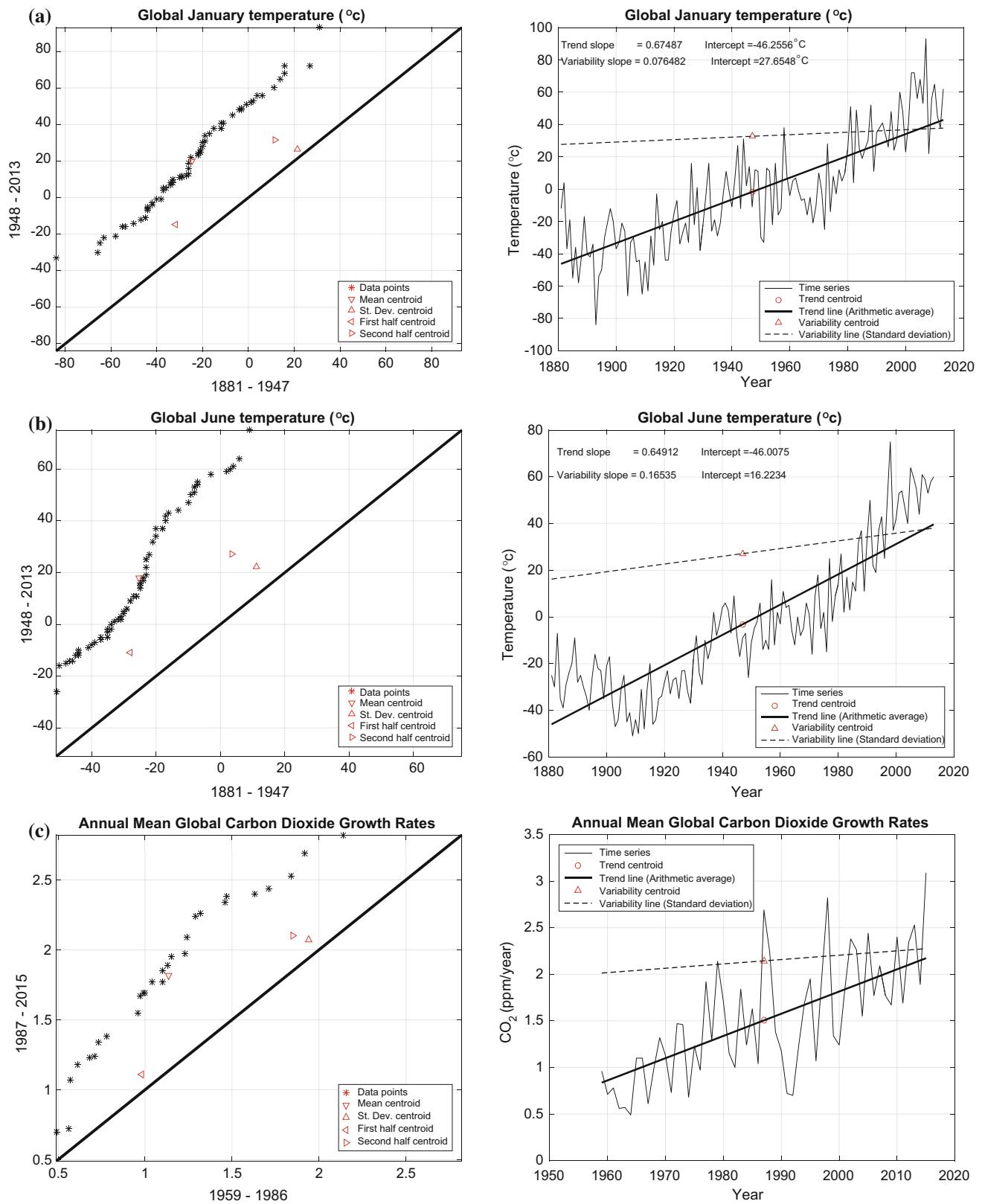
The following MATLAB program provides Spearman's rho test for trend possibility identification in a given time series.

Figure 10.4 presents various applications of the innovative trend template outcomes from the above MATLAB program application.

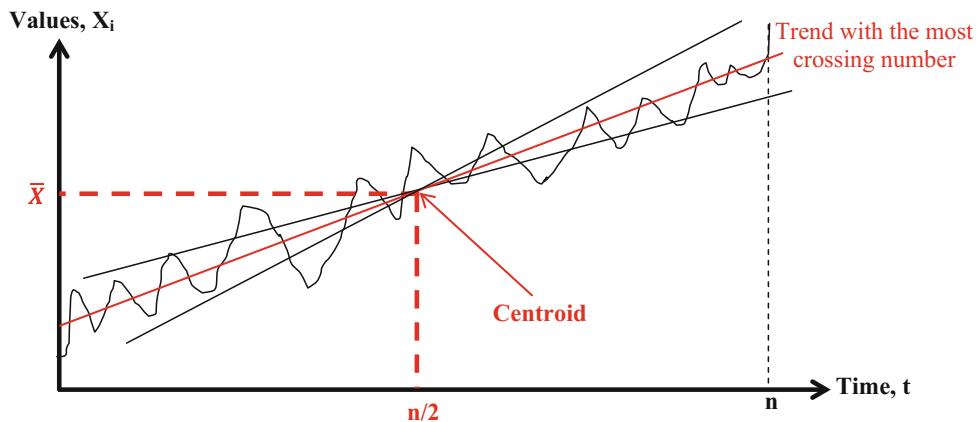
## 10.5 Crossing Trend Identification

Trend analyses are the necessary tools for depicting possible general increase or decrease in a given time series. There are many versions of trend identification methodologies such as the Mann-Kendall trend test, Spearman's tau, Sen's slope, regression line, and Şen's innovative trend analysis (Şen 2012, 2014). The literature has many papers about the use, cons and pros, and comparisons of various trend methodologies. In this section, a completely new approach is proposed based on the crossing properties of a time series. It is suggested that the suitable trend from the centroid of the given time series should have the maximum number of crossings (total number of up-crossings or down-crossings). This approach is applicable whether the time series has dependent or independent structure and also without any dependence on the type of the PDF. The validity of this method is presented through extensive Monte Carlo simulation technique and its comparison with other existing trend identification methodologies (Şen (2018)).

```
function [D,ZSR,Zcrit,p] = SpermanRho(X)
% This program is written by Zekai Sen on 26 October 2012
% This program calculates Sperman's Rho test
% X is the time series    NOTICE: ENTER TIME SERIES AS TRANPOSE OR NOT??
% NOTICE : SHOULD X BE ORDINARY OR TRANPOSE? CHECK WITH WRITING n
n1=size(X);
n=n1(1,1);
SortX=sort(X);
sum=0;
for i=1:n
    sum=sum+(SortX(i)-i)^2;
end
D=1-6*sum/(n*(n^2-1));
VD=1/(n-1); % D HAS NORMAL PDF WITH ZERO MEAN AND THIS VARIANCE
ZSR=abs(D/sqrt(VD));
Zcrit=normpdf(ZSR,0,1);
p=0.5-Zcrit;
end
```



**Fig. 10.4** Various ITT graphs: **a** global January temperature ( $^{\circ}\text{C}$ ), **b** global June temperature ( $^{\circ}\text{C}$ ), **c** annual CO<sub>2</sub> growth rate



**Fig. 10.5** Innovative crossing trend identifications

It is suggested that there are different number of crossings along various straight-lines that pass through the centroid of a given time series. The centroid has its abscissa at the half time duration ( $n/2$ ) and the ordinate equal to the arithmetic average of the given time series. The basis of the crossing trend methodology is to search the straight-line with the maximum number of crossings. Such a methodology is genuine and does not include any restrictive assumption.

This is a very recent trend identification methodology that is dependent on the crossing points along a straight-line as representative of possible trend that passes from the centroid of a given time series with a set of slope change. In this

manner the best representative monotonic trend can be caught after a series of trial-and-error procedure with increments in the slope value. In this procedure, the most important assumption is that the trend passes through the centroid point as shown in Fig. 10.5.

Along each candidate trend line the crossing number is either up-crossing or down-crossing of the summation of the two, after all the summation is twice of one of them. With this methodology the trend slope can be calculated automatically through the following MATLAB programs, which identifies the trend in a given time series based on the number of crossings.

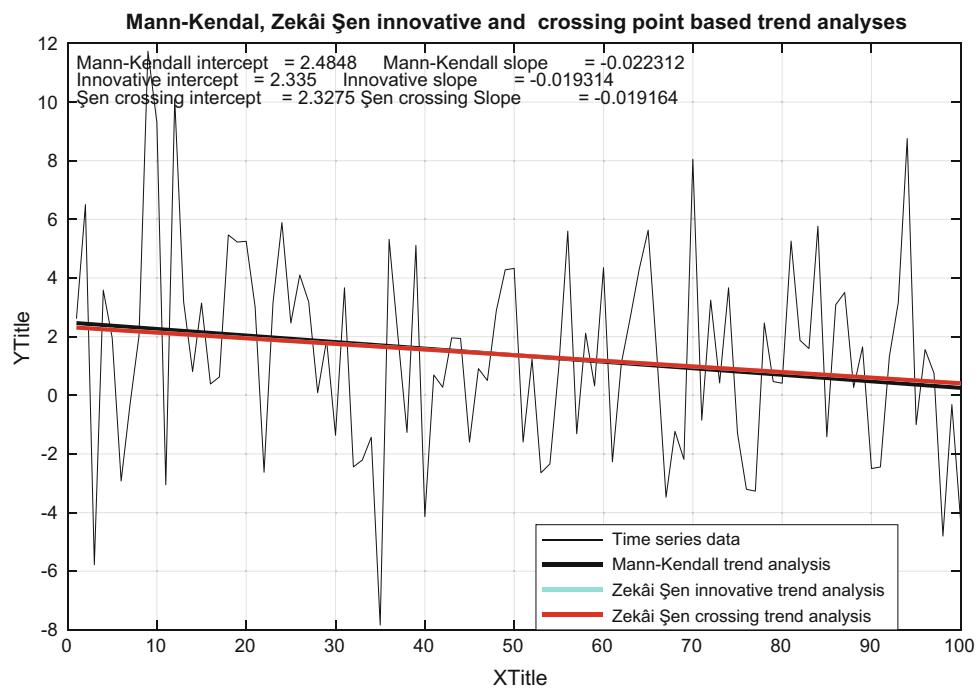
```

function [CN,Slope, Intercept] = InnovativeCrossingTrend(Y,SN,Title,Xtitle,Ytitle)
% This program is written by Zekai Sen on 21 August 2015
% This program finds the best trend in a time series by starting the slope
% from zero and increases it at equal increments based on SN steps
% SN      = Slope number
% Y       = Two dimensional array (TimeIntervalXDataValues)
% CN     = Trend crossing number
% S      = Incremental slope
% Slope   = Crossing trend slope
% Intercept = Crossing trend intercept
% Title   = Location or meteorology station description
% Xtitle  = Time series equal time intervals such as minute, day, month, year, etc.
% Ytitle  = Time series variable types such as precipitation, temperature, pressure, etc.
L=length(Y(:,2));
L2=L/2;
% Slope increment
ae(1)=0.00;
for i=2:SN
ae(i)=ae(i-1)+0.01;
end

```

.....Continued.....

```
% NEGATIVE SLOPE INCREMENTS
% Crossing number calculation
YM=mean(Y(:,2));
XM=L2;
for i=1:100
    s=ae(i);
    Yin=YM-s*L2;
    for j=1:L
        G(j)=Yin+s*(j-1); % Trend component
    end
    F=Y(:,2)-G'; % Time series trend difference
    UG=0; % Up-crossing number
    AG=0; % Down-crossing number
    k=0;
    for j=2:L
        if (F(j,1) >= 0 && F(j-1,1) < 0) % Up-crossing
            UG=UG+1;
            k=k+1;
            GN(k)=j-1+abs(F(j-1,1))/(abs(F(j-1,1))+abs(F(j,1))); % Up-crossing location number
        elseif (F(j,1) < 0 && F(j-1,1) >= 0) % Down crossing
            AG=AG+1;
            k=k+1;
            GN(k)=j-1+abs(F(j-1,1))/(abs(F(j-1,1))+abs(F(j,1)));
        end
    end
    CN(i)=UG+AG;
    BK(1)=GN(1)-1;
    NGN=length(GN);
    for j=2:NGN
        BK(j)=GN(j)-GN(j-1);
    end
    T(i)=sum(BK);
end
[GNSAY IG]=min(T); % High and small durations
SG=ae(IG);
[GSAY IM]=max(CN);
S=ae(IM);
% Sen's slope
k=0;
for i=1:L-1
    i1=i+1;
    for j=i1:L
        k=k+1;
        m(k)=(Y(j,2)-Y(i,2))/(j-i);
    end
end
Slope=median(m);
figure
plot(Y(:,1),Y(:,2),'k')
title>Title)
xlabel>Xtitle)
ylabel>Ytitle);
hold on
grid on
box on
Xmin=min(Y(:,1));
Xmax=max(Y(:,1));
Xmean=mean(Y(:,1));
Xin=mean(Y(:,2))-Slope*(Xmean-Xmin);
Intercept=Xin;
Xfi=mean(Y(:,2))+Slope*(Xmean-Xmin);
scatter(Xmean,mean(Y(:,2)), 'sr')
line([Xmin Xmax],[Xin Xfi], 'LineWidth',2) % Trend line
legend('Data values','Trend centroid','Trend line')
text(min(Y(:,1)),0.9*max(Y(:,2)),['Slope = ' num2str(Slope), ' Intercept = ' num2str(Intercept)]);
end
```



**Fig. 10.6** Crossing trend analysis procedure result

Figure 10.6 presents the crossing trend analysis application of the above MATLAB program to a time series with 100 records. In the same figure different trend identification methodology results are shown with their known names in the literature.

In support of the previous crossing trend MATLAB program its standard version is presented in the following box.

```

function [UG,CRS,MKS] = StandardInnovativeCrossingTrend(Y,Title,XTitle,YTitle)
% This program starts from the horizontal zero slope and increases the slope
% and finds the one that has the maximum crossing number, which
% corresponds to the trend line. (21 August 2015)
% ES = Slope number
% Y = TRend component
% UG = Upcrossing number
% CRS = Trend slope
% t = Test statistics
L=length(Y);
L2=L/2;
% Positive slope amounts
s(1)=1;
for i=2:10000
    s(i)=s(i-1)+0.0001;
end
YM=median(Y);
YS=std(Y);
y=(Y-YM)/YS;
for i=1:10000
    Yin=s(i)*L2;
    for j=1:L
        G(j)=Yin+s(i)*(j-1); % Trend component
    end
    F=y-G'; % Time series trend component difference
    UG=0; % Upcrossing number
    AG=0; % Downcrossing number
    k=0;
    for j=2:L
        if F(j,1) >= 0 & F(j-1,1) < 0 % Upcrossing
            UG=UG+1;
            k=k+1;
        else
            end
        end
    end
    [GSAY UGS]=max(UG);
    CRS=s(UGS); % Crossing slope
    % Sen's slope
    k=0;
    for i=1:L-1
        il=i+1;
        for j=il:L
            k=k+1;
            m(k)=(Y(j)-Y(i))/(j-i);
        end
    end
    figure
    plot(y,'k')
    hold on
    MKS=median(m); % Mann-Kendall Sen slope
    aMK=-MKS*L2; % Mann-Kendall intercept
    aCS=-CRS*L2; % Crossing innovative trend intercept
    x=1:1:L;
    yMK=aMK+MKS*x; % Mann-Kendall trend
    yCS=aCS+CRS*x; % Crossing innovative trend
    plot(x,yMK,'k','LineWidth',2)
    plot(x,yCS,'r','LineWidth',2)
    title>Title
    xlabel>XTitle
    ylabel>YTitle
    grid on
    box on
    legend('Time series data','Mann-Kendall trend analysis','Crossing innovative trend analysis')
    text(1,0.95*max(y),['Mann-Kendall intercept = ',num2str(aMK),' Mann-Kendall slope
    = ',num2str(MKS)]);
    text(1,0.90*max(y),['Sen crossing intercept = ',num2str(aCS),' Sen crossing Slope
    = ',num2str(CRS)]);
end

```

Figure 10.7 is the output from the above MATLAB program with the crossing trend component.

It is very obvious from this figure although it is possible to identify a holistic trend component, but there are partial trends

that cannot be represented by such a trend, and therefore, there is a need for partial trend analysis methodology, which is presented in Sect. 10.7. In the following box crossing trend simulation MATLAB program is given in full detail.

```

function InnovativeCrossingTrendSimulation(PDF,a,b,n,Rho,st,m,Title,XTitle,YTitle)
% This program simulates trend component for a given probability
% distribution function, sample length, correlation coefficient, and trend
% slope.
% PDF = Probability distribution function type
% a&b = PDF parameters
% n = Sample length
% Rho = Correlation coefficient
% st = Trend slope
% m = Ensemble number
n2=n/2;
n1=n-1;
for i=1:m
    if b == 0
        X=random(PDF,a,[n 1]); %Exponential independent values according to a given PDF
    else
        X=random(PDF,a,b,[n 1]); %Independent values according to a given PDF
    end
    XCor(1)=X(1);
    for j=2:n
        XCor(j)=Rho*XCor(j-1)+X(j)'; % Dependent values with correlation coefficient Rho
    end
    for j=1:n
        XCorTrend(j)=XCor(j)+st*j; % Dependent values with trend of slope st
    end
    YM=mean(XCorTrend);
    m1=mean(XCorTrend(1,1:n2));
    m2=mean(XCorTrend(1,n2+1:n));
    ms(i)=2*(m2-m1)/n; % Innovative slope slope
    s=random('normal',ms(i),5,[1000 1]);
    for j=1:1000
        Yin=YM-s(j)*n2;
        for k=1:n
            G(k)=Yin+s(j)*k; % Trend component
        end
        F=XCorTrend-G; % Time series trend component difference
        UG(j)=0; % Upcrossing number
        AG(j)=0; % Downcrossing number
        k=0;
        for k=2:n1
            if F(1,k) >= 0 && F(1,k-1) < 0 % Üste geçiş
                UG(j)=UG(j)+1;
                k=k+1;
                GN(k)=k-1+abs(F(1,k-1))/(abs(F(1,k-1))+abs(F(1,k))); % Upcrossing number
            elseif F(1,k) < 0 && F(1,k-1) > 0 % Alta geçiş
                AG(j)=AG(j)+1;
                k=k+1;
                GN(k)=k-1+abs(F(1,k-1))/(abs(F(1,k-1))+abs(F(1,k))); % Downcrossing
            else
                end
            end
            GS(j)=UG(j)+AG(j); % Crossing number
            BK(1)=GN(1)-1;
            NGN=length(GN); % Run lengths
            for j=2:NGN
                BK(j)=GN(j)-GN(j-1); % Run lengths
            end
            T(i)=sum(BK);
            [GSAY IM]=max(GS);
            CRSS(i)=s(IM); % Crossing simulation slopes
        end
    end
    CRS=mean(CRSS); % This can be adapted as mean, median, mode or maximum
    k=0;
    for i=1:n-1
        i1=i+1;
        for j=i1:n
            k=k+1;
            m(k)=(XCorTrend(j)-XCorTrend(i))/(j-i);
        end
    end
    MKS=median(m); % Mann-Kendal Sen slope
end

```

.....Continued.....

```

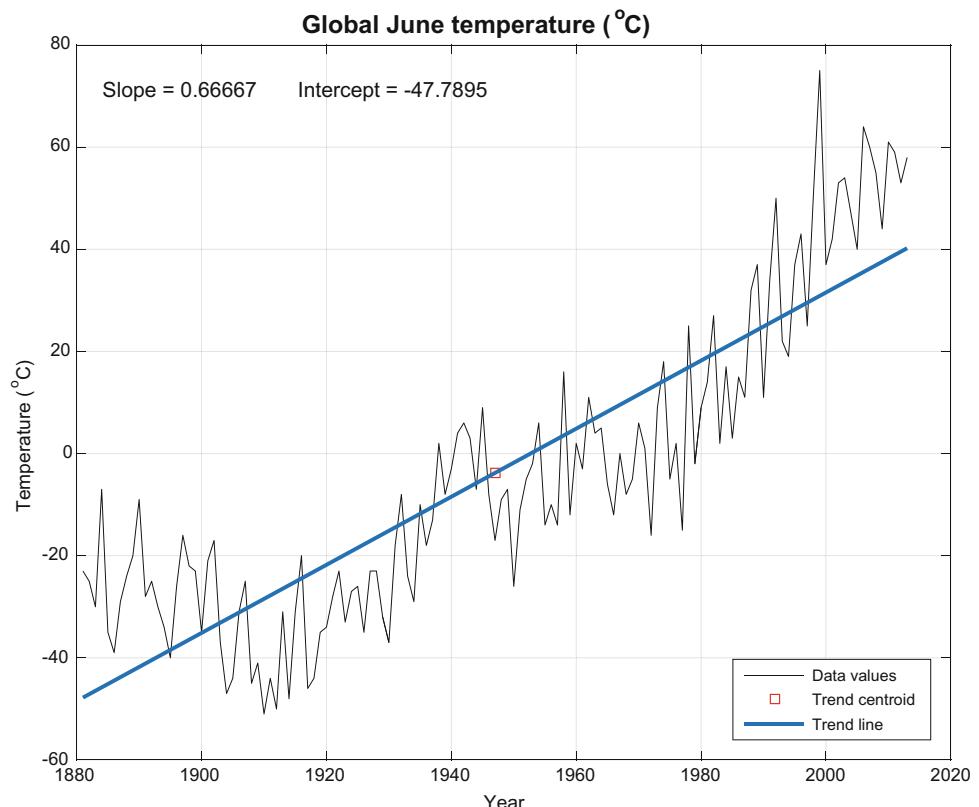
figure
plot(XCorTrend,'k')
hold on
aMK=YM-MKS*n2; % Mann-Kendall intercept
aZS=YM -mean(ms)*n2; % Zekai Şen innovative method intercept
aCS=YM-CRS*n2; % Crossing innovative trend intercept
x=1:1:n;
yMK=aMK+MKS*x; % Mann-Kendall trend
yZS=aZS+mean(ms)*x; % Zekai Şen innovative trend
yCS=aCS+CRS*x; % Crossing innovative trend
plot(x,yMK,'k','LineWidth',2)
plot(x,yZS,'c','LineWidth',2)
plot(x,yCS,'r','LineWidth',2)
title('Title')
xlabel(XTitle)
ylabel(YTitle)
grid on
box on
legend('Time series data','Mann-Kendall trend analysis','Zekai Şen innovative trend analysis','Zekai Şen crossing trend analysis')
text(1,0.98*max(XCorTrend),['Mann-Kendall intercept = ',num2str(aMK), ' Mann-Kendall
slope = ',num2str(MKS)]);
text(1,0.92*max(XCorTrend),['Innovative intercept = ',num2str(aZS), ' Innovative
slope = ',num2str(mean(ms))]);
text(1,0.86*max(XCorTrend),['Şen crossing intercept = ',num2str(aCS), ' Şen crossing
Slope = ',num2str(CRS)]);
text(1,0.80*max(XCorTrend),['Shape parameter = ',num2str(a)]);
text(1,0.74*max(XCorTrend),['Cor. coef. = ',num2str(Rho), '']);
text(1,0.74*max(XCorTrend),['Trend slope = ',num2str(st)]);
end

```

Figure 10.8 shows one of the ensemble components with its holistic trend component as a result of various trend analysis methodologies. Again one can observe that a holistic trend cannot depict the partial trend possibilities in the time series.

**Fig. 10.7** Holistic crossing trend components

The following MATLAB program provides a comparison of four different trend analysis methodologies, namely Mann-Kendall, ITT, regression, and crossing trend approaches.



```

function [GS,CRS,IM,GNSAY,SG,IG,s] =
MannKendalSenInnovativeRegressionAndCrossingTrends(Y,Title,XTitle,YTitle)
% Bu program sıfırdan başlamak üzere değişik eğimlerdeki geçiş sayısının en
% fazla olduğu durumda trendin eğimini tespit etmek için Zekai Şen
% tarafından 21 Agustos 2015 tarihinde (Cuma günü) yazılmıştır.
% ES = Eğim sayısı
% Y = Gidiş bileşeni araştırılan zaman serisi
% GS = Gidişte geçiş sayısı
% CRS = Gidiş eğimi
%
L=length(Y);
L2=L/2;
m1=mean(Y(1:L2,1));
m2=mean(Y(L2+1:L,1));
ms=2*(m2-m1)/L; % Mean slope
st=std(Y);
s=random('normal',ms,3,[1000 1]);
XM=L2;
YM=mean(Y);
for i=1:1000
    Yin=YM-s(i)*L2;
    for j=1:L
        G(j)=Yin+s(i)*(j); % Gidiş bileşeni
    end
    F=Y-G'; % Zaman serisi gidiş bileşeni farkı
    UG(i)=0; % Üst geçiş sayısı
    AG(i)=0; % Alt geçiş sayısı
    k=0;
    for j=2:L
        if F(j,1) >= 0 && F(j-1,1) < 0 % Üste geçiş
            UG(i)=UG(i)+1;
            k=k+1;
            GN(k)=j-1+abs(F(j-1,1))/(abs(F(j-1,1))+abs(F(j,1))); % Üste geçiş noktası
        elseif F(j,1) < 0 && F(j-1,1) > 0 % Alta geçiş
            AG(i)=AG(i)+1;
            k=k+1;
            GN(k)=j-1+abs(F(j-1,1))/(abs(F(j-1))+abs(F(j))); % Alta geçiş noktası
        else
        end
    end
    GS(i)=UG(i)+AG(i); % Geçi? say?s?
    BK(1)=GN(1)-1;
    NGN=length(GN); % Run lengths
    for j=2:NGN
        BK(j)=GN(j)-GN(j-1); % Run lengths
    end
    T(i)=sum(BK);
end
[GNSAY IG]=min(T); % Büyük ve küçük sürelerden
SG=s(IG);
[GSAY IM]=max(GS);
% TR=GS-GSAY;
% [TRmin ITR]=min(TR)
CRS=s(IM); % Crossing slope
% CRS=s(ITR); % Crossing slope
% Sen's slope
k=0;
for i=1:L-1
    il=i+1;
    for j=il:L
        k=k+1;
        m(k)=(Y(j)-Y(i))/(j-i);
    end
end
Xm=1;
% YM=max(Y);
% Regression trend analysis
X=1:1:L;
X=X';
mX=mean(X);
mY=mean(Y);
mXY=mean(X.*Y);
mX2=mean(X.*X);
M=[1 mX mX mX2];
PL=[mY mXY]/M; % Regression line intercept and slope calculation
.....Continued.....

```

```

figure
plot(Y,'k','LineWidth',2)
hold on
Sen=median(m); % Mann-Kendall Sen slope
aMK=YM-Sen*L2; % Mann-Kendall intercept
aZS=YM-ms*L2; % Zekai ?en innovative method intercept
aCS=YM-CRS*L2; % Crossing innovative trend intercept
yMK=aMK+Sen*X; % Mann-Kendall with Sen slope trend
yRL=PL(1)+PL(2)*X; % Regression trend
yZS=aZS+ms*X; % Zekai ?en innovative trend
yCS=aCS+CRS*X; % Crossing innovative trend
plot(X,yMK,'k','LineWidth',2)
plot(X,yRL,'r','LineWidth',2,'LineStyle','--')
plot(X,yZS,'c','LineWidth',2,'LineStyle',':')
plot(X,yCS,'m','LineWidth',2,'LineStyle','-.')
title>Title)
xlabel>XTitle)
ylabel>YTitle)
grid on
box on
legend('Time series data','Mann-Kendall trend analysis with Sen slope','Regression trend','Sen innovative trend analysis','Sen crossing trend analysis')
text(1,0.95*max(Y),['Mann-Kendall intercept = ',num2str(aMK),' Mann-Kendall slope
= ',num2str(Sen)]);
text(1,0.90*max(Y),['Regression intercept = ',num2str(PL(1)),', Regression slope
= ',num2str(PL(2))]);
text(1,0.85*max(Y),['Innovative intercept = ',num2str(aZS),', Innovative slope
= ',num2str(ms)]);
text(1,0.80*max(Y),['Sen crossing intercept = ',num2str(aCS),', Sen crossing Slope
= ',num2str(CRS)]);

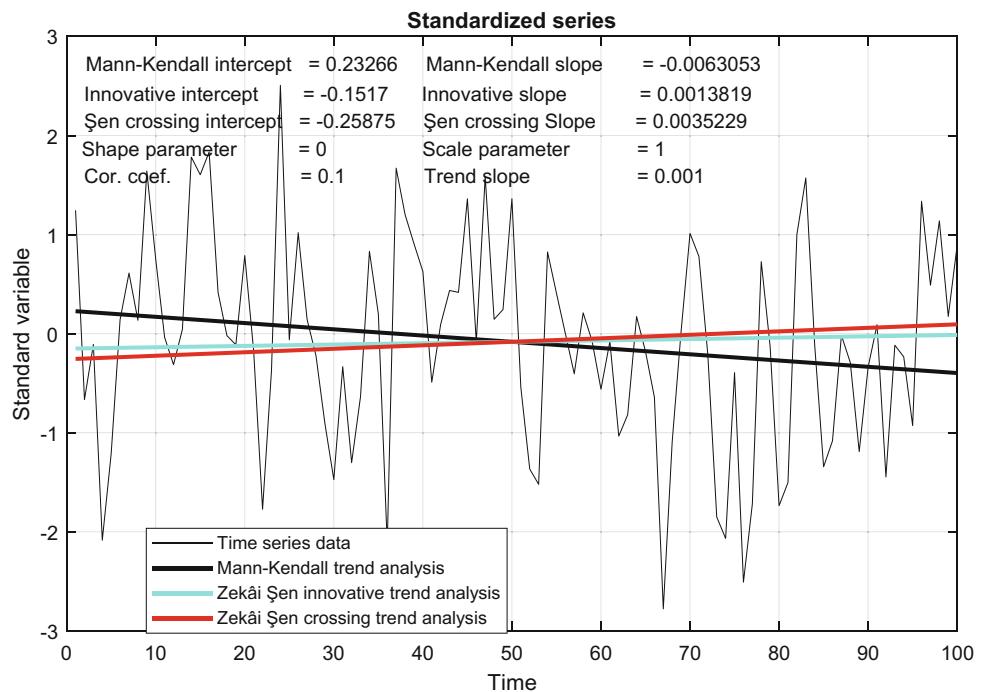
```

Figure 10.9 is an example that comes from the execution of the above MATLAB program to a standard time series.

It is obvious from this figure that the trend identification by the four methodologies falls onto each other. The main

reason is that the standard time series was adapted as serially independent and normally (Gaussian) PDF complying time series.

**Fig. 10.8** Crossing trend simulation result



## 10.6 Variation Trend

In the previous section, the trend is searched on the average along the time series duration, which provides information about systematic increase or decrease at the arithmetic average level. There are other statistical parameters of the same time series, and then the question is whether one can also look for any possible increase or decrease in the standard deviation? Such a search is referred to as the variation

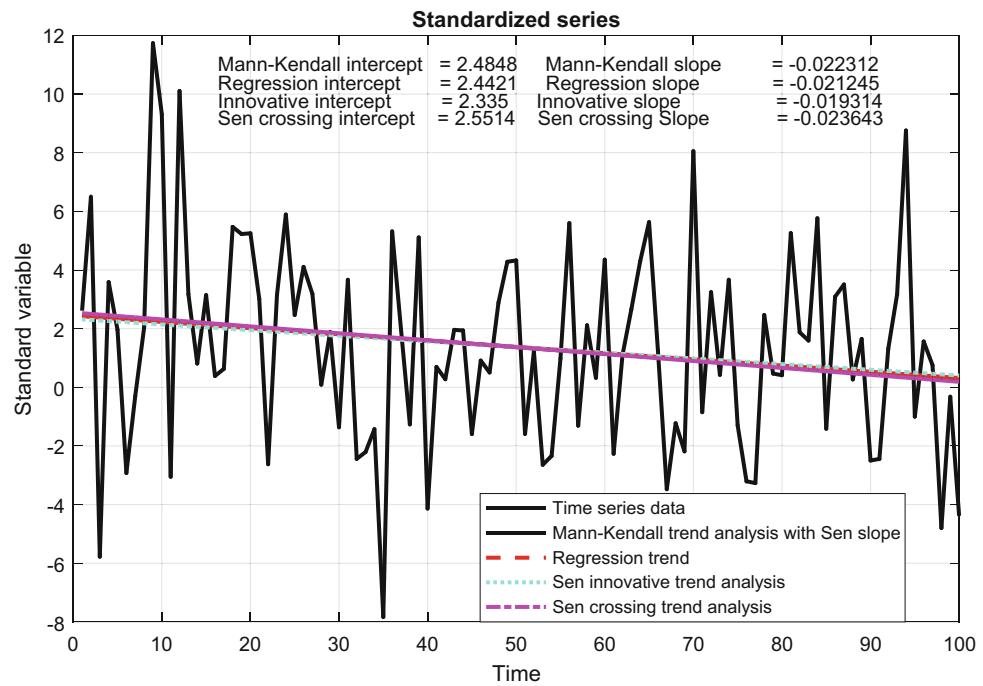
trend. Figure 10.2b implies also variation trends in terms of widening or narrowing of the standard deviation. It may even be possible to have no trend as for the arithmetic average is concerned but there may be trend on the fluctuations around the mean level as shown in Fig. 10.10a. There are other possibilities of trend in both the arithmetic average and the standard deviation (Fig. 10.10b, c).

The following MATLAB language written program is for variation trend identification in a given time series.

```

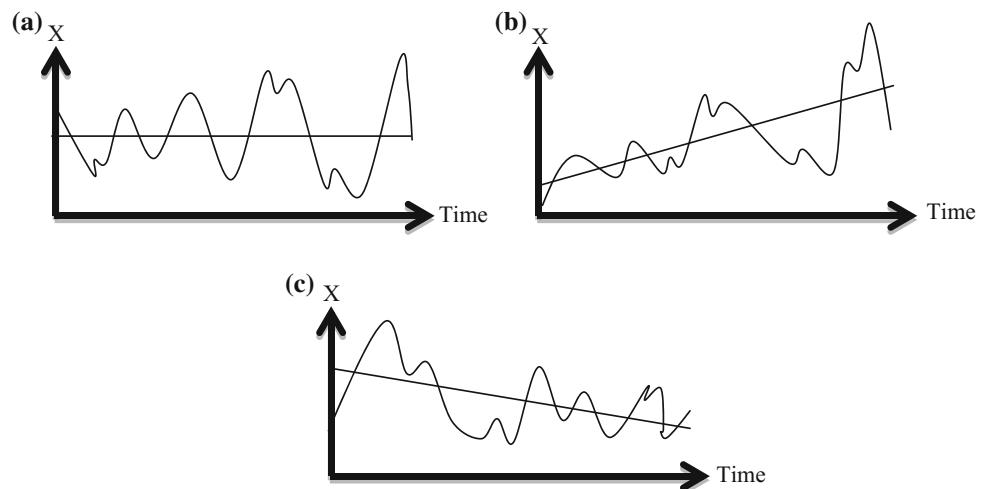
function [m1,m2,Sm,s1,s2,Ss,Sen] = TrendAndVariationPlots(X, Title, TitleX,TitleY)
% This program hs been written by Zekai Sen on 1 Temmuz 2015 Çarşamba gece
% saat 01:13
% X is a data vector of two columns the first column includes years and
% the second data values
n=length(X);
Xm=mean(X(:,2));
n2=n/2;
figure
plot(X(:,1),X(:,2), 'k')
hold on
title(Title)
xlabel(TitleX)
ylabel(TitleY)
grid on
box on
m1=mean(X(1:n2,2));
m2=mean(X(n2+1:n,2));
Sm=2*(m2-m1)/n;
am=Xm-Sm*n2;
Xminm=min(X(:,1));
Xmaxm=max(X(:,1));
Xim=am+Sm*1;
Xfm=am+Sm*n;
line([Xminm Xmaxm],[Xim Xfm], 'LineWidth',1,'Color','k','LineStyle', '-')
s1=std(X(1:n2,2));
s2=std(X(n2+1:n,2));
Ss=2*(s2-s1)/n;
as=Xm-Ss*n2;
xis=as+Ss*min(X(:,1));
xfs=as+Ss*max(X(:,1));
Xis=am+Ss*1;
Xfs=am+Ss*n;
line([Xminm Xmaxm],[Xis Xfs], 'LineWidth',1,'Color','k','LineStyle', '--')
[Sen] = SenSlope(X(:,2));
asen=Xm-Sen*n2;
xisen=asen+Sen*1;
xfsen=asen+Sen*n;
line([Xminm Xmaxm],[xisen xfsen], 'LineWidth',1,'Color','k','LineStyle', '-.')
Ymax=max(X(:,2));
text(Xminm+1,Ymax-1,[' Trend slope = ' num2str(Sm), ' Variability slope = ' num2str(Ss), ' Sen
slope = ' num2str(Sen)])
legend('Temperature (mm)', 'Trend line', 'Variability line', 'MK line', 'Location', 'Northeast')
end

```



**Fig. 10.9** Four different types of trend methodological application results

**Fig. 10.10** Variation trends:  
**a** standard deviation, **b** arithmetic average and standard deviation increase trends, **c** arithmetic average and standard deviation decreasing trends



The following subprogram is called by the above MATLAB program for the completion of the trend and variability components.

conceptually possible to have a series of partial trend components within a time series as increasing and decreasing

```
function [Slope] = SenSlope(X)
% This program is written by Zekai Sen on August 2012 (Thursday)
% It calculates only Sen slope
% X is the given time series
% THE FOLLOWING STEPS ARE FOR Sen's SLOPE CALCULATION
n=length(X);
k=0;
for i=1:n-1
    for j=i+1:n
        k=k+1;
        m(k)=(X(j)-X(i))/(j-i);
    end
end
Slope=median(m);
end
```

The output of the variation trend identification MATLAB program is presented in Fig. 10.11 with the classical trend component.

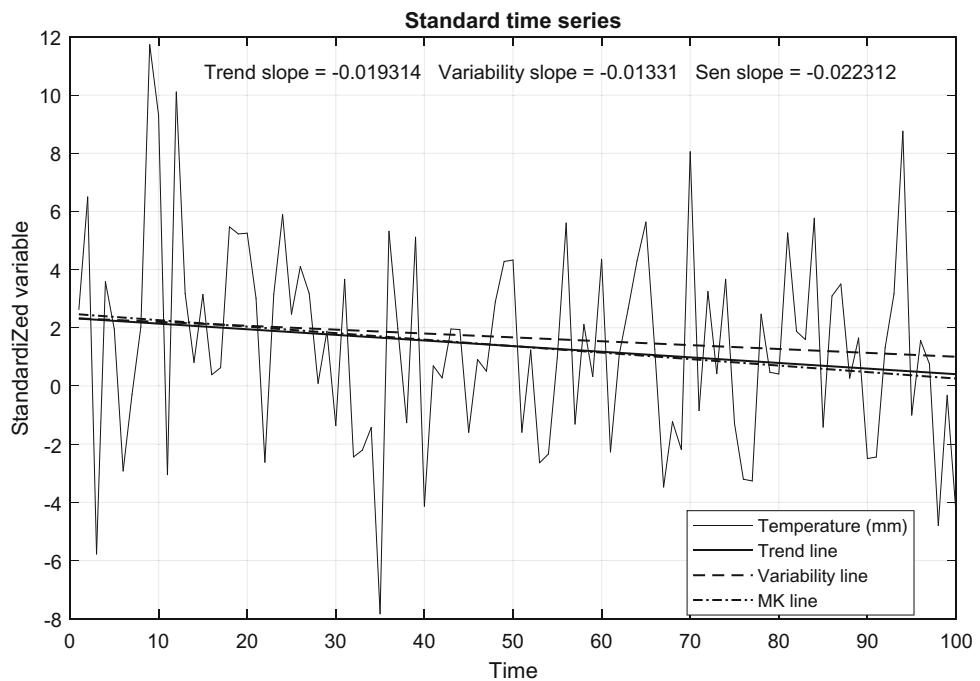
sequences. Hypothetically, Fig. 10.12 provides few samples for partial trend cases.

The following trend program in MATLAB language provides partial trends of any desired duration, such as 5, 10, 25, in a given time series.

## 10.7 Partial Trend Components

So far in all the previous sections monoclinic and holistic trends are discussed in a given series on the arithmetic average or standard deviation parameter levels. It is

**Fig. 10.11** Variation trend component in a given standard and independent time series



```

function PartialInnovativeTrendSequence(X,SD,Title)
% On 25 December 2015
% This program calculates 10-yearly trends based on the first 10-year data
% X = Time series
% SD = Sub-duration: For instance, 10, 15,20,25,... year
% NOTICE: THIS PROGRAM IS VALID FOR 100-YEAR TIME SERIES AT THE MAXIMUM
N=length(X); % Total length of time series
NDT=floor(N/SD); % Number of decadal trends
NDT1=NDT-1; % Possible pair of sequences
xor=SD; %Because the length of time series considered will be SD time units
figure
plot(X,'k')
box on;
grid on;
title(Title)
xlabel('Time')
ylabel('Rainfall (mm)')
hold on
% Sequence of successive sub-durations
for i=1:NDT1
    OncekiAltSure=sort(X(NDT1+1:i*SD));
    SonrakiAltSure=sort(X(i*SD+1:(i+1)*SD));
    xdif=SonrakiAltSure-OncekiAltSure; % The difference between the first and the last
parts
    td=mean(xdif); % Deviation from the non-trend line
    b(i)=2*td/(2*SD); % Trend slope
    XX=X(NDT1+1:i*SD);
    yor=mean(XX); % Arithmetic average of given time series
    a(i)=yor-b(i)*xor;
    xi=i*SD;
    yi=a(i);
    xf=(i+1)*SD;
    yf=a(i)+b(i)*SD;
    line([xi xf],[yi yf], 'LineWidth',2, 'Color','r');
end
text(15,max(X)-(max(X)-min(X))*0.05,['Intercept: a= ' num2str(a(1:NDT1))])
text(15,max(X)-(max(X)-min(X))*0.1,['Slope: b= ' num2str(b(1:NDT1))])
end

```

For a given time series, this MATLAB program produces successive partial trend components as in Fig. 10.13.

## 10.8 Fuzzy Trend Analysis

There are many natural earth systems events that evolve by time, and the major question is whether they have more or less statistically similar random fluctuations around a mean level or are there intervals (clusters) with systematic increases or decreases on the average level? In general, almost everybody thinks to compare (during his/her life the natural events such as temperature, rainfall, snow) the present-day behaviors with their childhood youngster ages so as to reach a conclusion whether there is a significant change in terms of increase or decrease linguistically. Such a verbal reasoning is based on approximate reasoning and information is rather vague, imprecise, uncertain, and incomplete, which are collectively expressed by a single word “fuzzy” (Zadeh 1999, 2001). However, in any fuzzy

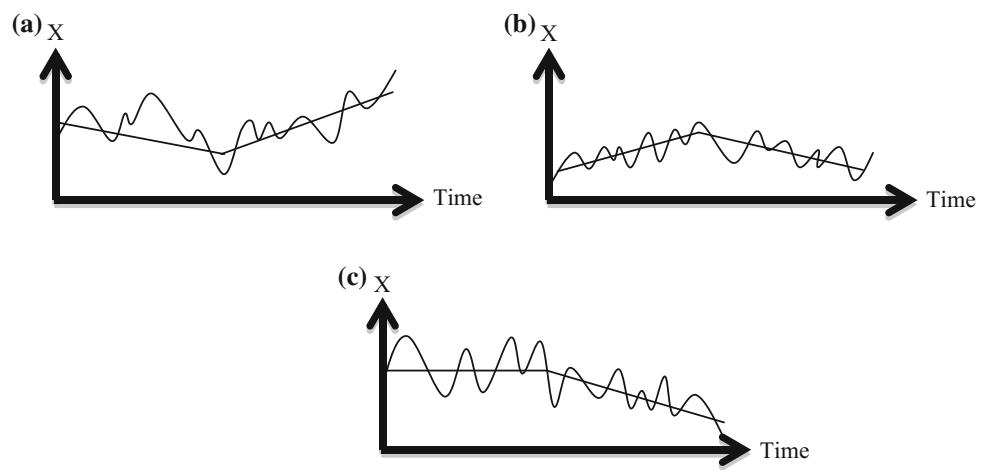
reasoning there are hidden quantitative facts that need for their deductions by numerical methodologies. In such reasoning, most often absolute time intervals are not considered, but rather relativistic time periods are in the comparison. Such reasoning and comparisons do not lead to holistic but to piecewise features. The reasoning is based on experience and expert views.

The basis of the approach rests on the same concept as presented in Sect. 10.3 for the ITT. In this section, a given time series is standardized prior to possible trend identification. If the given time series,  $X_i$  ( $i = 1, 2, \dots, n$ ), has  $n$  measurements then its standardized sequence can be obtained as,

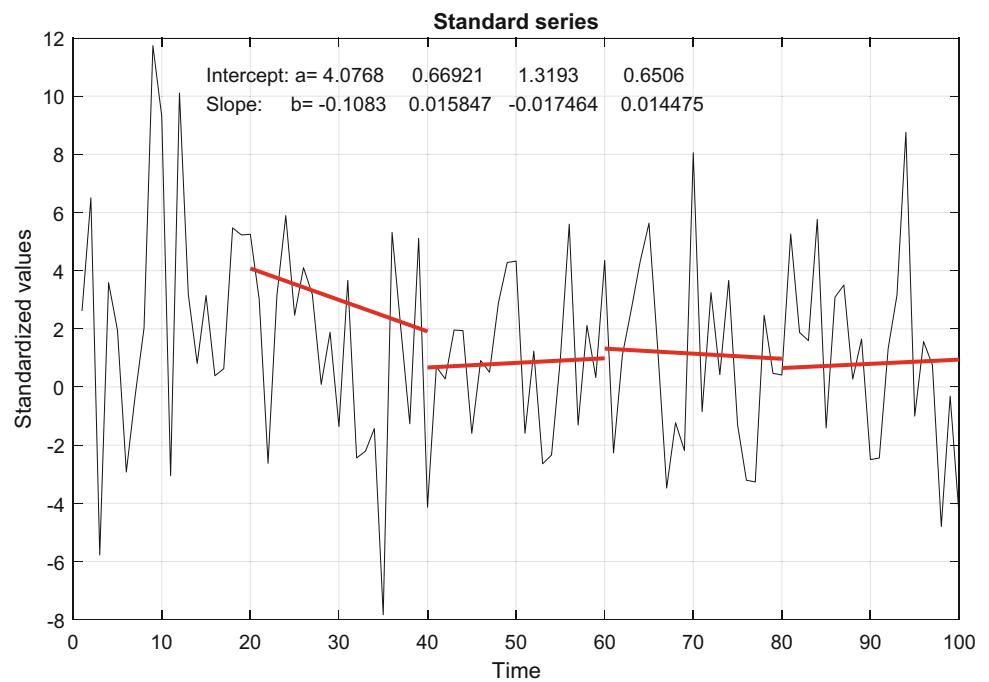
$$x_i = \frac{X_i - \bar{X}}{S_X} \quad (10.5)$$

where  $\bar{X}$  and  $S_X$  are the arithmetic mean and standard deviation, respectively. This transformation results in a standard dimensionless series with zero mean and unit standard deviation.

**Fig. 10.12** Partial trend components



**Fig. 10.13** Partial trend components

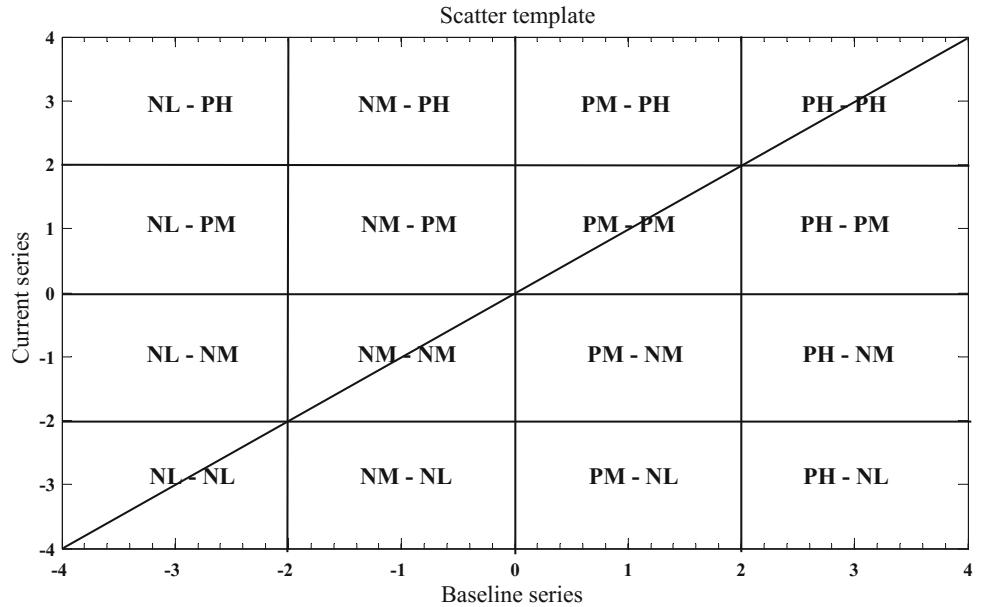


In the fuzzy trend search identification procedure, a certain number of sub-series are compared with some other non-overlapping sub-series of the same length. The necessary basis for such an approach has been given briefly by Sen (2012). Such an approach provides the ability for trend identification possibilities with respect to the baseline. The best template for such a comparison is the Cartesian coordinate system box as in Fig. 10.14.

According to this figure, the variation domain is divided into 16 squares each with special symbols and verbal (linguistic) specification as in Table 10.1.

Such a partition provides the following important interpretation facilities for comparison of standardized time series.

- (1) One can make verbal interpretations about mutual characteristics for any given pair of base and current series values,
- (2) It is possible to classify verbally a group of data according to their sub-area position,
- (3) Different groups (classes) of data can be compared as to their neighborship or relative position to each other and again verbal interpretations are possible,

**Fig. 10.14** Scatter template**Table 10.1** Sub-area verbal specifications<sup>1</sup>

Number	Symbol	Verbal specification
1	NL-NL	Negative low–Negative low
2	NM-NL	Negative medium–Negative low
3	PM-NL	Positive medium–Negative low
4	PH-NL	Positive high–Negative low
5	NL-NM	Negative low–Negative medium
6	NM-NM	Negative medium–Negative medium
7	PM-NM	Positive medium–Negative medium
8	PH-NM	Positive high–Negative medium
9	NL-PM	Negative low–Positive medium
10	NM-PM	Negative medium–Positive medium
11	PM-PM	Positive medium–Positive medium
12	PH-PM	Positive high–Positive medium
13	NL-PH	Negative low–Positive high
14	NM-PH	Negative medium–Positive high
15	PM-PH	Positive medium–Positive high
16	PH-PH	Positive high–Positive high

- (4) If the scatter points appear along the NL-NL, NM-NM, PM-PM, and PH-PH sub-areas, then the two different time series or sub-series from the same time series are closely similar to each other. If they are parts of the same data, for instance, halves, then this implies that there is not trend component within the series,
- (5) If the scatter of points appears along the NL-PH, NM-PM, PM-NM, and PH-NL sub-areas, then these

two series are negatively related to each other or there are decreasing trends in the time series.

If the scatter of points indicates a parallel line to  $45^\circ$  line in the increasing and decreasing regions of Fig. 10.5, then there is a monochromatic trend. Otherwise, any curvy appearance of the scatter points implies nonlinear or broken trend existence (Sen 2017). The relevant software is given in the MATLAB programming language as in the following box.

```

function [LowL, MidL, HigL] = LowMiddleHighTrends(X, Title, XTitle1, YTitle1, XTitle2, YTitle2)
% This program is written by Zekai Sen on 25 June 2015
% X          = Two dimensional array (TimeIntervalsxDataValue)
% Title      = The location or station name
% XTitle1    = The time duration of the first half such as 1881-1947
% XTitle2    = Time series equal time interval duration such as minute, day, month, year,
etc.
% YTitle1    = The time duration of the first half such as 1947-2013
% YTitle2    = The variable type such as precipitation, temperature, etc.
% LowL       = Between low value limit between the lowest data and minus one standard
deviation from the mean
% MidM       = Between minus and plus one standard deviation from the mean
% HigL       = Between plus one standard deviation from the mean and the maximum data value
Y=X(:,2);
n=length(X(:,2)); % Time series length (Data number)
n2=floor(n/2); % Half data number
Xm=mean(Y); % Time series mean value
Tm=mean(X(:,1)); % Time mean
FH=sort(X(1:n2,2)); % First half
SH=sort(X(n2+1:2*n2,2));
FHm=mean(FH);
FHS=std(FH);
SHm=mean(SH);
SHs=std(SH);
SlopeX=2*(SHm-FHm)/n;
InterX=Xm-SlopeX*Tm;
figure
scatter(FH, SH, 'k.')
hold on
grid on
box on
title(Title)
xlabel(XTitle1)
ylabel(YTitle1)
scatter(FHm, SHm, 'r*')
Xmax=max(max(FH),max(SH));
Xmin=min(min(FH),min(SH));
xlim([Xmin Xmax])
ylim([Xmin Xmax])
line([Xmin Xmax], [Xmin Xmax], 'LineWidth', 2)
% Calculate low, medium and high intervals
Int=(Xmax-Xmin)/3;
LowL=Xmin+Int; % Low upper limit
MidL=LowL+Int; % Middle upper limit and high lower limit
HigL=max(Y);
% Lower data values
Low=Y(Y<=LowL);
[FL IL]=sort(Low); % Find the sequence of lower limit at the First half
NLD=max(IL); % Number of low data
NLD2=floor(NLD/2);
LowDF=Low(1:1:NLD2); % Low data first half values
LowDS=Low(NLD2:1:2*NLD2); % Low data second half values
LowDFm=mean(LowDF); % Low data first half mean
LowDSm=mean(LowDS); % Low data second half mean
LowDFs=std(LowDF); % Low data first half standard deviation
LowDSS=std(LowDS); % Low data second half standard deviation
scatter(LowDFm, LowDSm, 'rs')
% Middle data values
Mid=Y(Y>LowL&Y<MidL); % Middle data values
[Mid IM]=sort(Mid); % Find the sequence of middle limit at the First half
NMD=max(IM); % Number of low data
NMD2=floor(NMD/2);
MiddleDF=Mid(1:1:NMD2); % Middle data first half values
MiddleDS=Mid(NMD2:1:2*NMD2); % Middle data second half values
MiddleDFm=mean(MiddleDF); % Middle data first half mean
MiddleDSm=mean(MiddleDS); % Middle data second half mean
MiddleDFs=std(MiddleDF); % Middle data first half standard deviation
MiddleDSS=std(MiddleDS); % Middle data second half standard deviation
scatter(MiddleDFm, MiddleDSm, 'rh')
% High data values
High=Y(Y>MidL&Y<HigL); % High data values
[Hig IH]=sort(High); % Find the sequence of middle limit at the First half
NHD=max(IH); % Number of low data
NHD2=floor(NHD/2);
HighDF=High(1:1:NHD2); % High data first half values
HighDS=High(NHD2:1:2*NHD2); % High data second half values
continued.....

```

```

HighDFm=mean(HighDF); % High data first half mean
HighDSm=mean(HighDS); % High data second half mean
HighDFs=std(HighDF); % High data first half standard deviation
HighDSs=std(HighDS); % High data second half standard deviation
scatter(HighDFm,HighDSm,'rd')
legend('Location','Southeast','Data values','Mean centroid','No-Trend line','Low
centroid','Middle centroid','High centroid')
% LOW, MIDDLE AND HIGH VALUES DETERMINATION IN TIME SERIES SEQUENCE
% Slopes and intercepts
[S I]=sort(X(:,2));
% Low data calculations
XL=S(S<LowL);
Ll=length(XL);
XLF=XL(1:Ll/2); % Low data first half
XLS=XL(Ll/2+1:Ll); % Low data second half
SXL=2*(mean(XLS)-mean(XLF))/n; % Low data slope
IL=I(1:Ll);
ILm=mean(IL);
IXL=mean(XL)-SXL*ILm; % Low data intercept
XLin=IXL+SXL*1; % Low data initial value at n=1
XLfi=IXL+SXL*n; % Low data initial value at n = number of data
% Medium data calculations
XM=S(S>LowL&S<=MidL);
Lm=length(XM);
XMF=XM(1:Lm/2); % Medium data first half
XMS=XM(Lm/2+1:Lm); % Medium data second half
SXM=2*(mean(XMS)-mean(XMF))/n; % Medium data slope
IM=I(Ll+1:Ll+Lm);
IMm=mean(IM);
IXM=mean(XM)-SXM*IMm; % Medium data intercept
XMin=IXM+SXM*1;
XMfi=IXM+SXM*n;
% High data calculations
XH=S(S>MidL);
Lh=length(XH);
XHF=XH(1:Lh/2); % High data first half
XHS=XH(Lh/2+1:Lh); % High data second half
SXH=2*(mean(XHS)-mean(XHF))/n; % High data slope
IH=I(Ll+Lm+1:end);
IHm=mean(IH);
IXH=mean(XH)-SXH*IHm; % High data intercept
XHin=IXH+SXH*1;
XHfi=IXH+SXH*n;
figure
Tin=min(X(:,1)); % Monotonic trend initial time value
Xin=InterX+SlopeX*Tin; % Monotonic trend time series initial value
Tfi=max(X(:,1)); % Monotonic trend final time value
xlim([Tin Tfi])
Xfi=InterX+SlopeX*Tfi; % Monotonic trend time series final value
plot(X(:,1),X(:,2),'k')
hold on
grid on
box on
title>Title)
xlabel>XTitle2)
ylabel>YTitle2)
line([Tin Tfi],[Xin Xfi],'LineWidth',2,'Color','m')
text(2,0.9*Xmax-1,['M. Slope a = ' num2str(SlopeX),' L. Slope a = ' num2str(SXL),' H. Slope a = ' num2str(SXH)]);
text(2,0.8*Xmax-5,['M. Intercept b = ' num2str(InterX),' L. Intercept b = ' num2str(IXL),' H. Intercept b = ' num2str(IXH)]);
plot(min(X(:,1))+IL,XL,'cs')
plot(min(X(:,1))+IM,XM,'md')
plot(min(X(:,1))+IH,XH,'r*')
line([Tin Tfi],[XLin XLfi],'LineWidth',2,'Color','c') % Low trend line
line([Tin Tfi],[XHin XHfi],'LineWidth',2,'Color','r') % High trend line
legend('Time series','Trend line','Low values','Medium values','High values','Low
trend','High trend','Location','Southeast')
end

```

Figure 10.15 presents a set of fuzzy trend component analysis graphs from different parts of Turkey.

In the following box, the MATLAB program provides “low,” “medium,” and “high” monotonic trend components.

```

function LowMediumHighTrendsPlots(X,dim)
% This program is written by Zekai Sen on 9 August 2012 for distinguishing
% between low, medium and high trends
% dim is the number of column for ascending order sorting
NoOfData=length(X);
scatter(1:11:20,X(:,dim));
lsline
hold on
[xsort,xindex]=sortrows(X,dim);
s=size(X);
n=s(1,1);
n3=round(n/3-0.5);
plot(xsort(1:n3,1),xsort(1:n3,dim),'*g')
lsline
plot(xsort(n3+1:2*n3,1),xsort(n3+1:2*n3,dim),'*y')
lsline
plot(xsort(2*n3+1:3*n3,1),xsort(2*n3+1:3*n3,dim),'*r')
lsline
end

```

The application of this program to global warming temperature January records is presented in Fig. 10.16 according to the ITT procedure, and in Fig. 10.17 monotonic trends are generated by means of the last MATLAB program for “low,” “medium,” and “high” categories.

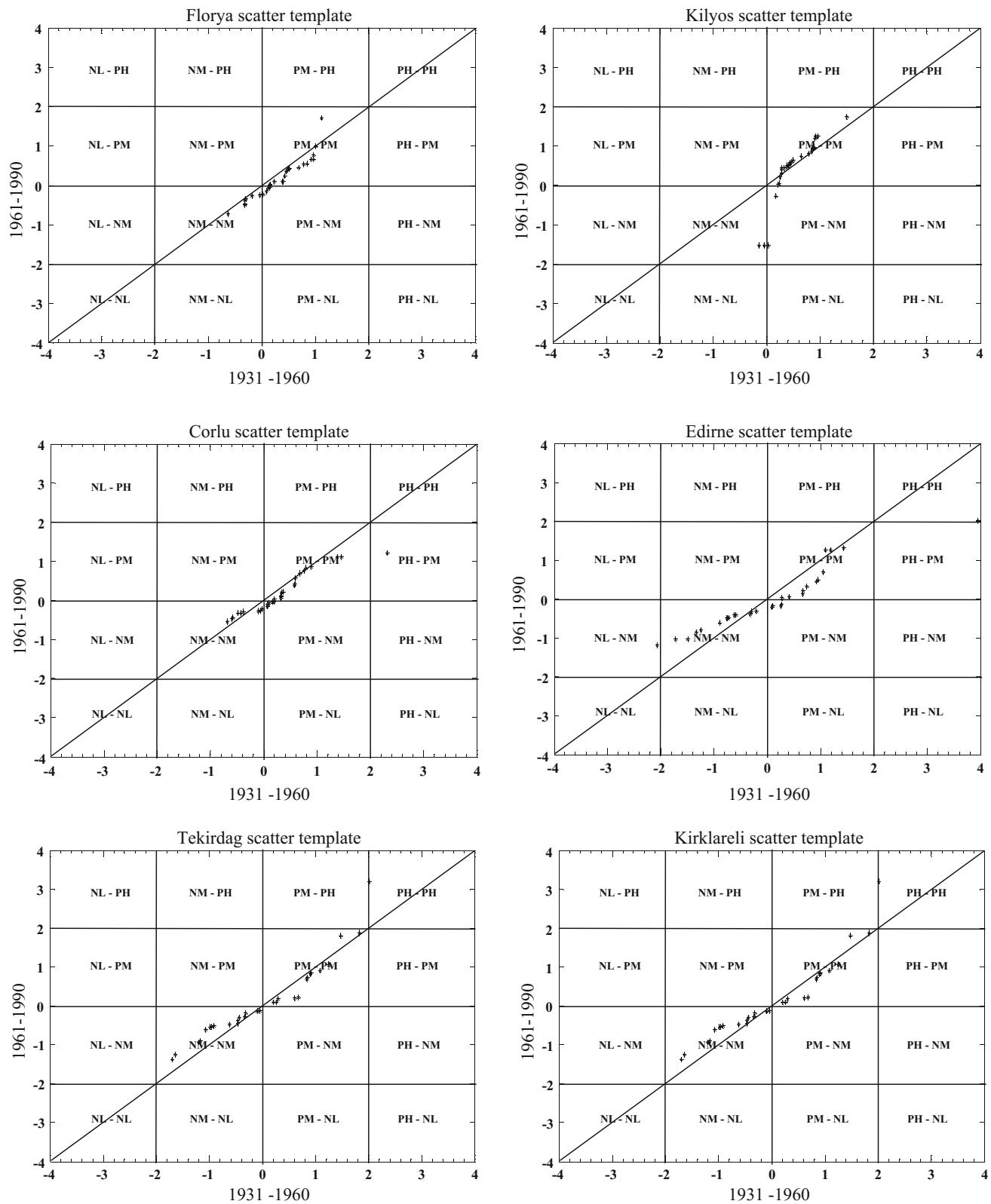
## 10.9 Over-whitening Trend Analysis

In, the previous sections’ different methodologies are employed for trend component identification in time series. Some of them have a set of assumptions for the validity, but in practical applications they are not cared for always. The classical Mann-Kendal trend test requires that the time series should have serially independent structure. To render any serially dependent structure into independent structure, a pre-whitening (PW) procedure has been suggested (Zhang et al. 2001; Hamilton et al. 2001; Burn and Hag Elnur 2002). Yue et al. (2002) have shown that PW procedure is successful only in the case of first-order autoregressive, AR(1), process with an increasing absolute trend and positive serial correlation coefficient only. In this

section, instead of PW, over-whitening (OW) procedure is suggested for the same purpose by Sen (2016) through extensive analytical formulations, which are not repeated here. Analytically necessary formulations for OW are presented with ITT assessment procedure through extensive simulation studies.

To expose the validity of the OW procedure in trend identification, two simulation sets are presented—one with first-order serial correlation coefficients and the trend slopes yielding the same template without OW and the other for OW time series with different OW standard deviations. A series of Monte Carlo computer simulations are performed with a set of statistical parameters using AR(1) process as already explained in Chap. 6. The standard time series (zero mean and unit variance) are generated for the simulation procedure based on the set of autocorrelation coefficients ( $\rho = \pm 0.1, \pm 0.3, \pm 0.5, \pm 0.7, \text{ and } \pm 0.9$ ) and the trend slopes set ( $s = \pm 0.001, \pm 0.003, \pm 0.005, \pm 0.007, \text{ and } \pm 0.009$ ) with length 1000. The MATLAB program below is for the OW simulation work.

The following MATLAB programs provide the necessary simulation works for OW trend procedure application.



**Fig. 10.15** Fuzzy trend component templates

```

function [slope,m1,m2,xdif] = SingleTrendSimulationWhitening(r,s,ave,std,a,N)
% This program generates lag-on Markov process trend simulation and then
% adds white noise process on the top of the time series so as to apply
% prewhitening procedure(written by Zekai Sen on 26 May 2013
% r = corelation coefficient
% s = trend slope
% ave = average
% std = standard deviation
% a =whitening ratio (Find it from prewhitening chart, Sen, 2013 No need
% for prewhitening in trend analysis (unpublished article)
% N = number of data
ran=random('normal',ave,std,[N,1]);
X(1)=ave+std*ran(1);
for j=2:N % Lag-one Markov process generation
    X(j)=ave+r*(X(j-1)-0)+std*sqrt(1-r^2)*ran(j);
end
for l=1:N % Trend incorporation
    XT(l)=X(l)+s*l; % i for correlation coefficient; j for trend slope;
    %and l for incorporation
end
% Prewhitening calculations
gam=sqrt(1/a-1); % This is the standard deviation of prewhitening process
WhiteRan=random('normal',ave,gam,[N,1]);
for l=1:N % White noise component incorporation
    XTW(l)= XT(l)+ WhiteRan(l);
end
figure
plot(XTW(1:end))
box on
grid on
xlabel('Number of data')
ylabel('Whitened time series')
M=max(XTW);
text(3,M-3,['Gamma = ',num2str(a), ' Tend slope = ',num2str(s), ' Correlation coeff. =',
num2str(r)])
title('Pre-whitened Markov process')
figure
X=sort(XTW(1,1:N/2));
Y=sort(XTW(1,(N/2+1):N));
scatter(X,Y,'k*')
hold on
xlabel('Lag-one Markov process first half')
ylabel('Lag-one Markov process second half')
box on
grid on
axis equal
m=min(X,Y);
M=max(X,Y);
line([m M],[m M],'LineWidth',2)
% title('Correlation coefficient = ',r,';trend slope = ', s)
m1=mean(sort(XTW(1,1:N/2)));
m2=mean(sort(XTW(1,(N/2+1):N)));
scatter(m1,m2,'r')
xdif=sort(XTW(1,1:N/2))-sort(XTW(1,(N/2+1):N));
figure
plot(xdif(1:end))
box on
grid on
xlabel('Half duration')
ylabel('Differences')
title('Difference between the first and second halves')
slope=mean(xdif)/(N/2);
end

```

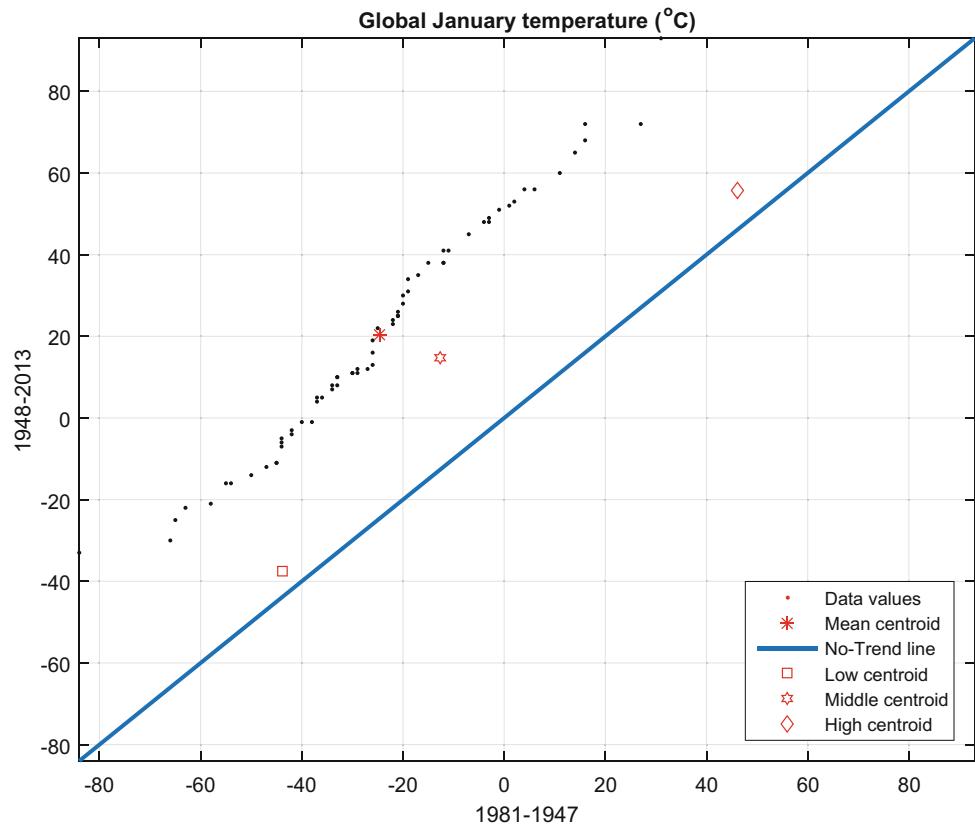
After the application of the simulation procedure by means of the above MATLAB program run, Fig. 10.18 presents one of the synthetic ITT results with OW procedure.

The sequence is embedded with a trend component of slope equal to 0.003. The resulting time series is shown in Fig. 10.19a with innovative trend template in Fig. 10.19b. Addition of the component over the same sequence and then the application of the innovative trend procedure yield the

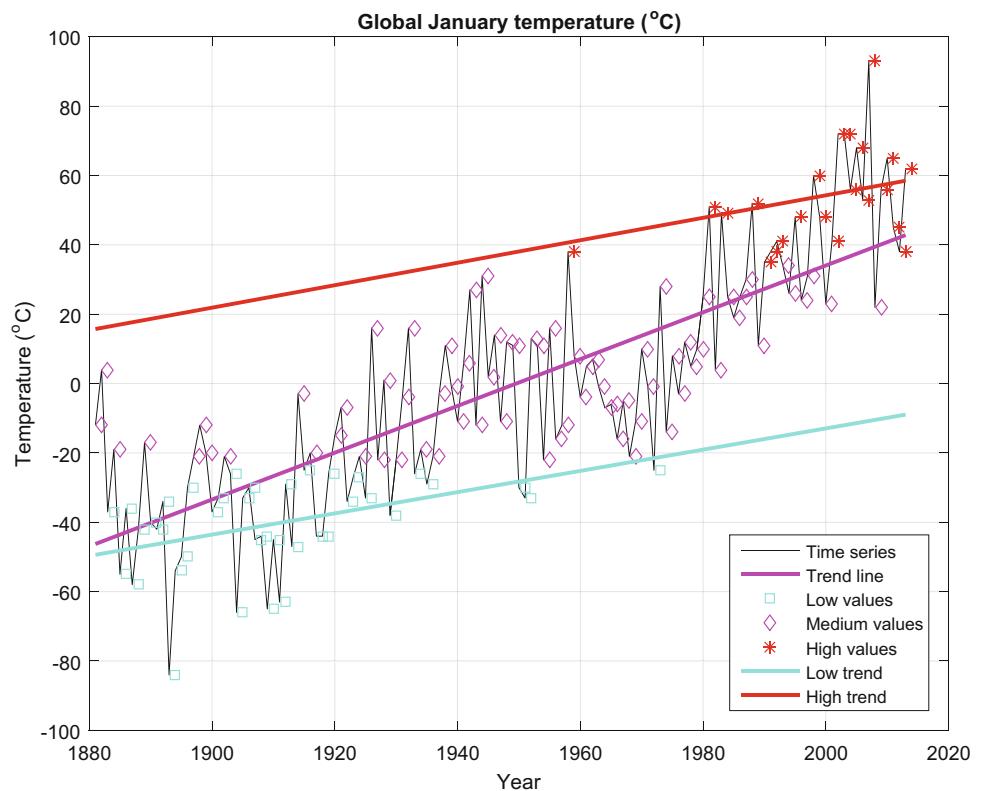
trend template as in Fig. 10.19c, which is scaled down version of Fig. 10.19b. Figure 10.19b, c is overlapped, and the result, in Fig. 10.19d, shows whether over-whitened or not the same trend is preserved even in the over-whitened series.

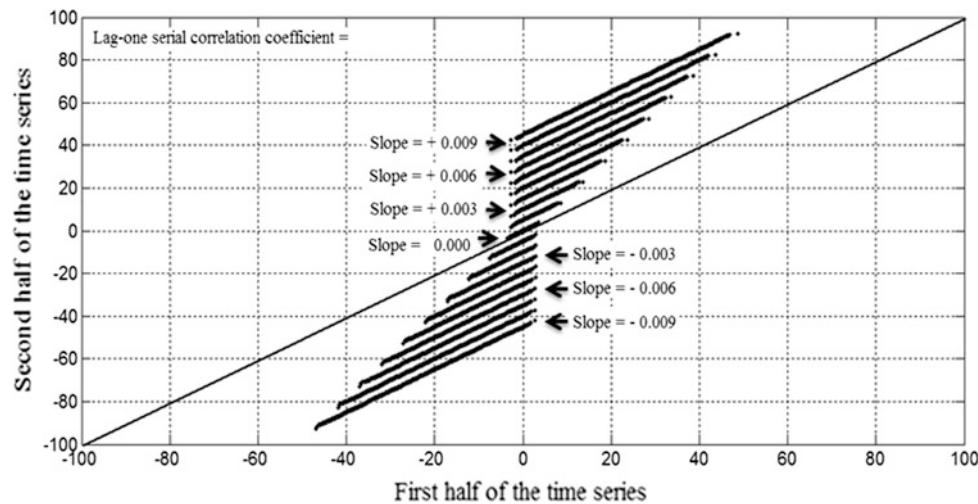
The application of the OW procedure to actual time series is presented for New Jersey annual temperature records for more than 100 years of data. The results are presented in Fig. 10.20 after the OW procedure implementation.

**Fig. 10.16** ITT procedure trend component templates

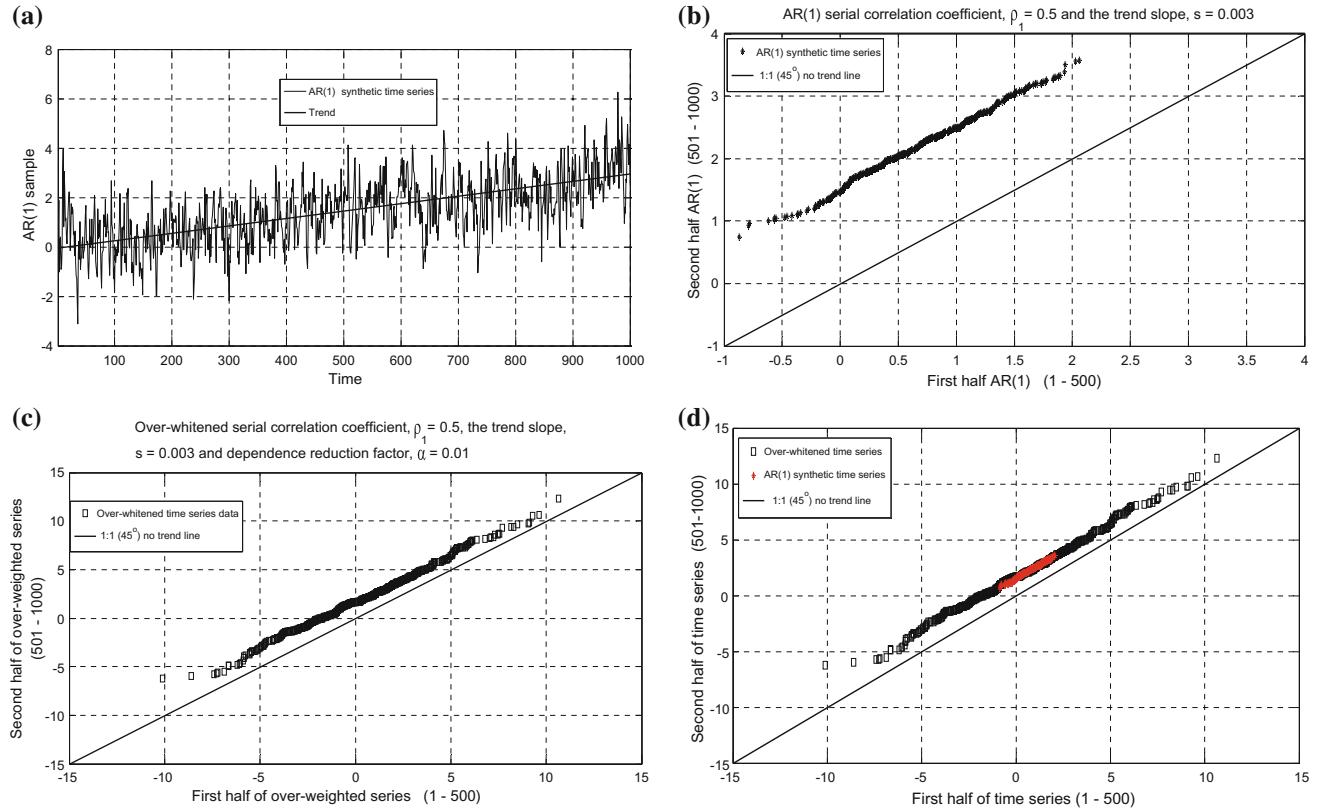


**Fig. 10.17** “Low,” “medium,” and “high” trend components



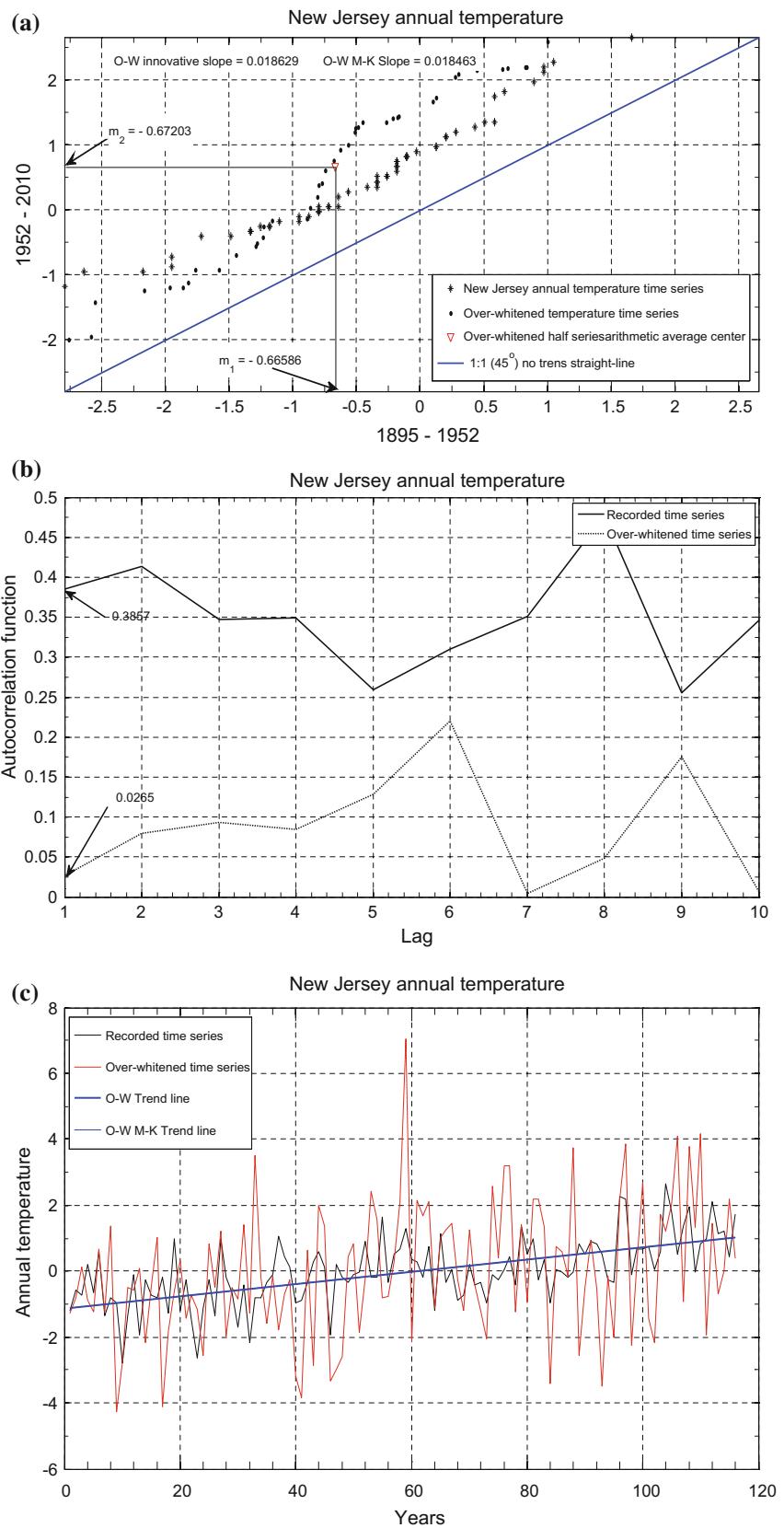


**Fig. 10.18** General innovative template for trend slopes irrespective of serial correlation coefficient

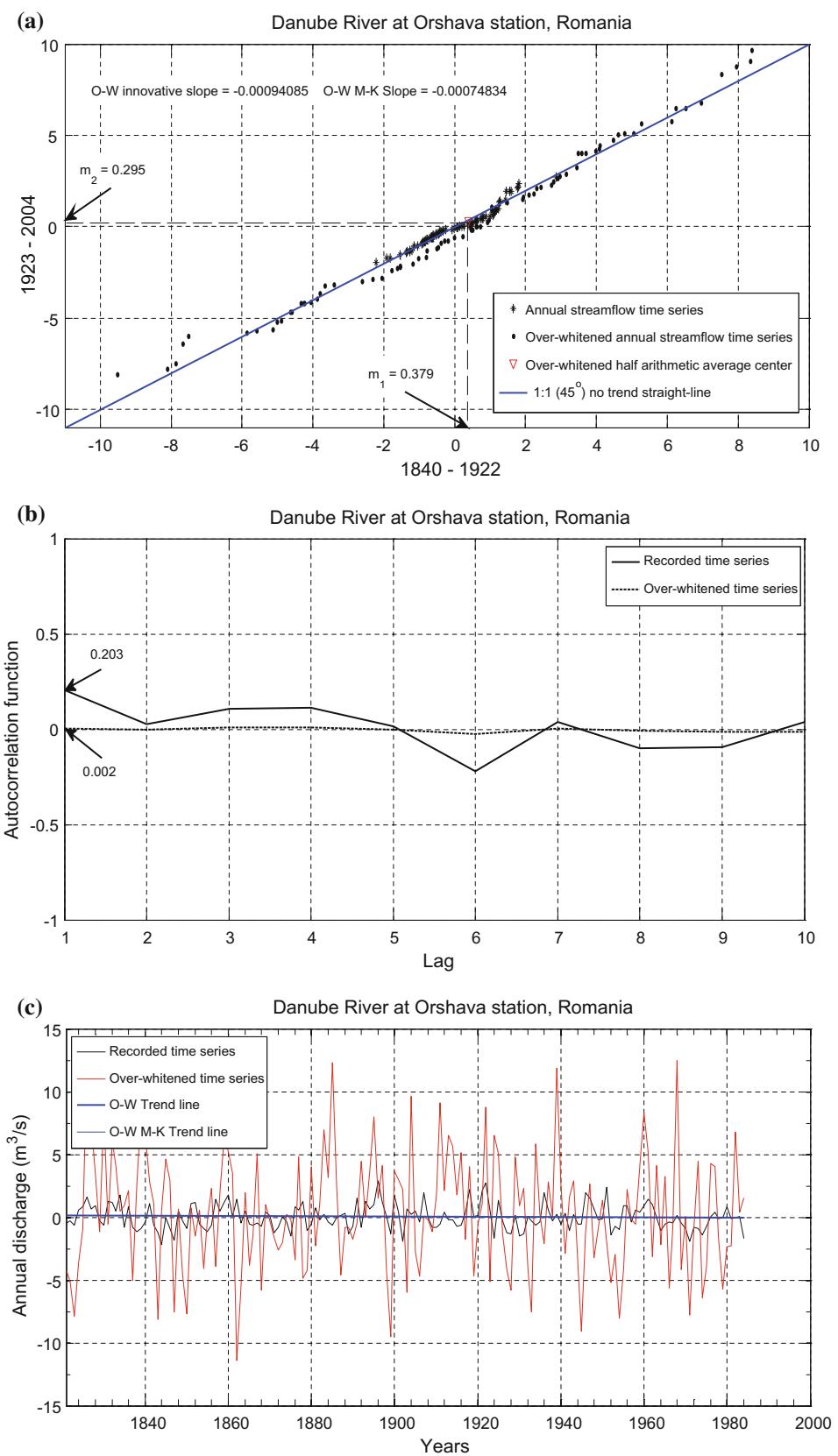


**Fig. 10.19** **a** Trend embedded time series, **b** trend prior to OW, **c** trend posterior to OW, **d** prior to and posterior to OW

**Fig. 10.20** New Jersey annual temperature OW procedure applications, **a** standardized and OW time series innovative template, **b** autocorrelation graph, **c** time series and trend graph



**Fig. 10.21** Danube River discharge OW procedure applications, **a** standardized and OW time series innovative template, **b** autocorrelation graph, **c** time series and trend graph



Another application is presented in Fig. 10.21, again for more than 100 years of record from Danube River.

## References

- Box, G.E.P., and Jenkins, G.M., (1970). Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco.
- Burn, D.H., Hag Elnur, M.A., (2002). Detection of hydrologic trends and variability. *J. Hydrol.*, Vol. 255: 107–122.
- Cox, D.R., and Miller, H.D., (1965). The Theory of Stochastic Processes Methuen, London.
- Douglas, E.M., Vogel, R.M., Kroll, C.N., (2000). Trends in floods and low flows in the United States: impact of spatial correlation. *Journal of Hydrology*, Vol. 240: 90–105.
- Faticchi, S., Barbosa, S.M., Caporali, E., and Silva, M.E., (2009). Deterministic versus stochastic trends: Detection and challenges. *Journal of Geophysical Research*, Vol. 114: D18121–D18132.
- Hamilton, J.P., Whitelaw, G.S., and Fenech, A., (2001). Mean annual temperature and annual precipitation trends at Canadian biosphere reserves, *Environ. Monit. Assess.*, Vol. 67: 239–275.
- IPCC (2007), “Working group II contribution to the intergovernmental panel on climate change fourth assessment report climate change 2007: Climate change impacts, adaptation and vulnerability”, p. 9–10.
- IPCC, (2013). Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1535 pp.
- Kendall, M.G., (1975). Rank Correlation Methods, Charles Griffin, London.
- Mann, H.B., (1945). Nonparametric tests against trend. *Econometrica*, Vol. 13(3): 245–259.
- Sen, P. K., (1968). Estimates of the regression coefficient based on Kendall’s Tau. *J. Am. Stat. Assoc.*, Vol. 63: 1379–1389.
- Sen, Z., (2012). Innovative trend analysis methodology, *Journal of Hydrologic Engineering*, Vol. 17(9): 1042–1046.
- Sen, Z., (2014). Trend Identification Simulation and Application, ASCE, *Journal of Hydrologic Engineering*, Vol. 19(3): 635–642.
- Sen, Z., (2015a). Innovative trend significance test and applications, *Theoretical and Applied Climatology*, <https://doi.org/10.1007/s00704-015-1681-x>.
- Sen, Z., (2015b). Innovative Trend methodologies in Science and Engineering. Springer, 349 pp.
- Sen, Z., (2016). Hydrological trend analysis with innovative and over-whitening procedures. *Hydrological Sciences Bulletin*, Vol. 62 (2): 294–305.
- Sen, Z., (2017). Global warming quantification by innovative trend template method. *Int. J. Global Warming*, Vol. 12(3–4): 499–512.
- Sen, Z., (2018). Crossing trend analysis methodology and application for Turkish rainfall records, *Theoretical and Applied Climatology*, Vol. 131(1–2): 285–293.
- Von Storch, V.H., (1995). Misuses of statistical analysis in climate research. In *Analysis of Climate Variability: Applications of Statistical Techniques*, von Storch H, Navarra A (Eds). Springer-Verlag: Berlin, 11–26.
- Yue, S., and Wang, C.Y., (2004). The Mann-Kendall Test Modified by Effective Sample Size to Detect Trend in Serially Correlated Hydrological Series. *Water Resources Management*, Vol. 18: 201–218.
- Yue, S., et al., (2002). The influence of autocorrelation on the ability to detect trend in hydrological series. *Hydrological Processes*, Vol. 16: 1807–1829.
- Zadeh, L. A., (1999). From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, 45:105–119.
- Zadeh, L.A., (2001). The problem of deduction in an environment of imprecision, uncertainty, and partial truth. In: Nikravesh M, Azvine B (eds), FLINT 2001, New Directions in Enhancing the Power of the Internet, UC Berkeley Electronics Research Laboratory, Memorandum No. UCB/ERL M01/28.
- Zhang, X., Harvey, K.D., Hogg, W.D., Yuzyk, T.R., (2001). Trends in Canadian streamflow. *Water Resour. Res.*, Vol. 37(4): 987–998.

# Index

## A

- Actual evapotranspiration, 37
- Angström method, 217–222, 224, 225, 229, 232
- Anions, 115, 116, 118–122, 126, 127, 133, 136, 137
- Approximate
  - reasoning, 2, 4, 262
- Aquifer
  - parameters, 89–91, 93, 95, 98, 99
  - philosophy, 90
  - test, 6, 90–93, 98, 101, 106
- ARIMA model, 150, 154–157, 180
- Aspect, 1, 2, 4–6, 53, 54, 78, 181, 229
- Association matrix, 136
- Atomic weight, 116

## B

- Bank-full, 54, 55, 67

## C

- Cations, 115, 116, 119–122, 125–127, 130, 133, 136, 137
- Circular diagram, 120, 121
- Climate change, 5–7, 16, 33, 35, 155, 171–175, 180, 195, 217, 243, 244
- Climate modeling, 172, 189
- Climate variability, 171
- Computation, 1, 4–6, 12, 137, 199
- Computer
  - program, 4–6
- Confined, 78, 89, 91, 92, 95, 101, 102, 107, 113
- Crossing trend, 249, 251, 253, 254, 256, 258
- Cross-section, 54–57, 67, 69, 72, 73, 111
- Cumulative semi-variogram, 197, 199–201, 207, 211, 232

## D

- Digital
  - computers, 1, 2, 5
- Digital elevation, 6, 53
- Dimensionless
  - intensity, 15, 16
- Direct flow, 36, 38
- Directional semi-variogram, 202, 205, 206, 208
- Discharge, 53–57, 67, 69, 72–74, 85, 90, 95, 110–113, 239, 240, 273
- Downscaling
  - dynamic, 174, 176
  - principles, 174

## A

- statistical, 155, 172, 174, 175, 181

- Drought
  - index, 80
  - model, 6, 7, 43, 53, 172, 175, 178, 189
- Dry periods, 78, 79, 172

## E

- Efficiency factor, 35
- Electric conductivity, 115, 117
- Energy tree, 239, 240
- Equivalent per million, 116

## F

- Flood
  - flash, 10, 78, 155
  - risk, 72
  - warning, 67, 72
- Fossil fuels, 171, 217, 229
- Fracture, 91, 106–111
- Frequency factor, 31, 33
- Fuzzy classification, 137
- Fuzzy trend, 263, 266, 268

## G

- General circulation model, 7, 155, 171–175, 180–182, 187
- Gradient, 53, 54, 89, 95, 111, 193, 241
- Groundwater
  - management, 7, 110–112
  - optimization, 110
- H
- Harmonic analysis, 46, 50, 145
- Hydrochemistry, 6, 115, 137
- Hydrogeology, 89, 95
- Hydrograph
  - unit, 53, 69, 73
- Hydrology, 53, 54, 67, 69, 72, 74, 80, 89, 172
- Hydropower, 217, 239, 240

## I

- Independent process, 145–147, 179, 180
- Innovative trend
  - template, 245, 247, 249, 269

- variability, 247
- Intensity  
duration, 7, 9, 12, 15  
frequency, 7, 9, 12
- Ions  
concentration, 115–120, 130, 131, 133, 134, 136, 138  
units, 115
- K**  
Karstic, 89, 91, 92  
Known seasonality, 50
- L**  
Large diameter, 89, 96, 97, 99, 103–105
- Leakage factor, 101–103
- Leaky, 89, 91, 92, 101–107
- Logic  
crisp, 4  
fuzzy, 4, 137, 139  
rules, 1–6
- M**  
Markov model, 146–153, 155, 165, 179, 180
- Mathematics, 2, 3
- Maximum rainfall, 13, 15, 21, 31, 33–35, 73, 155
- Medium  
fractured, 91, 92  
karstic, 91, 92  
porous, 91, 92, 106, 107
- Meteorology, 7, 12, 14–16, 23, 28, 32–35, 37, 42, 80, 161, 171–175, 181, 187, 193, 229, 233
- Monotonic trend, 243–245, 251, 266, 267
- Multi-rectangular diagram, 126, 131
- N**  
Newtonian energy principle, 240
- Nonlinear model, 223
- Nugget, 201
- O**  
Over-whitening trend, 267
- P**  
Partial trend, 7, 243, 254, 256, 261–263
- Parts per million, 115
- Percentage error, 116
- Philosophy, 1–4, 90
- Piezometric level, 91, 101
- Point  
cumulative semi-variogram, 200
- Porous, 89, 91, 92, 106, 107
- Potential evapotranspiration, 37, 172
- Power formulation, 232
- Precipitation  
maximum, 30, 31
- Probabilistic innovative model, 225
- Probability  
distribution function, 9, 43, 143–146, 157, 173, 188, 196, 223, 225, 244
- Probable maximum  
flood, 73, 74
- Probably  
maximum, 34  
precipitation, 127
- Pumping test, 91
- R**  
Radar  
data, 21
- Radius of influence, 112, 113, 176, 202, 211
- Rainfall  
maximum, 13, 15, 21, 22, 31, 33–35, 73, 155  
annual, 12, 13, 15, 21, 22, 73, 155  
annual—daily, 12, 13, 15, 21, 22, 73, 155
- Rainfall efficiency, 36
- Range, 22, 36, 175, 196, 237
- Rating curve, 55, 56, 72
- Regional climate model, 172, 174–176
- Regional estimation, 176, 209, 210, 214
- Renewable  
energy, 217, 239
- Risk, 5–7, 9, 10, 12, 15, 16, 53, 72, 74, 75, 77, 79, 143, 145, 161, 175, 243
- Risk calculation, 74, 75
- S**  
Saturation, 89–91, 95, 101, 111
- Scenario, 155, 171–175, 180, 181, 187–189
- Schoeller diagram, 120
- Science  
philosophy, 2–4
- Seasonal generation, 164
- Semi-variogram, 197, 199, 202, 204–207, 211
- Sill, 196, 201, 202, 204
- Similarity model, 134
- Simulation, 4–6, 143–146, 155, 157, 161, 163–165, 167, 169, 174, 175, 244, 245, 249, 254, 258, 267, 269
- Slope, 10, 53–55, 89, 91, 98–101, 143, 161, 220, 221, 225, 243–245, 247, 249, 251, 267, 269, 271
- Slope method, 98
- Snow  
storage, 36, 172
- Sodium adsorption ratio, 115, 119
- Software  
philosophy, 3
- Soil moisture  
capacity, 36  
storage, 36
- Solar energy, 217, 220, 221, 223–226
- Spatial  
analysis, 193
- Spatial dependence  
function, 16, 175–177, 193, 194, 207
- Spatial dependence function, 16, 175–178, 181, 182, 193–195, 201, 207, 209–211
- Standard ion index template, 130

Standard precipitation index, 43, 44, 46, 47, 80

Statistical downscaling

software, 180

Stiff diagram, 119–121

Stochastic processes, 6, 143, 179

Storativity, 89, 98, 106

Storm

rainfall, 7–10, 12, 13, 16, 53, 239

Storyline, 172, 173

Successive ion index, 131, 133

Successive substitution model, 221

Sunshine duration, 37, 217–221, 223–226, 229, 231

Surplus flow, 36

## T

Technology, 1, 2, 5, 171, 180

Temperature, 7, 22, 36, 37, 115, 118, 171–175, 181, 193, 218, 244, 250, 262, 267, 269, 272

Temporal model, 179–181, 187

Thalweg, 54

Thronthwaite, 37

Total

dissolved solids, 115–118

hardness, 118

Transmissivity, 89–91, 93, 95, 98, 101, 111, 112

Trend concept, 243

Trend slope, 161, 243–245, 247, 251, 267, 271

Trend variability, 247

Triangular diagram

Piper, 121

Turbine design, 234, 235, 237, 239

Type curves

Hantush, 101

Theis, 91, 94–99, 101

## U

Unconfined, 89–92, 101

Unit hydrograph

dimensionless, 69

Unrestricted model, 220

Unsaturation, 90

Upscaling, 171

## V

Variation trend, 259–261

Variogram, 195, 196

Void, 4, 89, 110, 111, 119

## W

Water balance, 36, 37, 42, 45

Water quality index, 133, 134, 136

Well, 2, 3, 78, 89–92, 95–99, 101–107, 110–113, 133, 193, 232, 240, 241, 243

Wet periods, 79

White Markov Process (WMP), 148, 152, 153, 155, 175, 179

Wind energy, 229, 232, 233, 235–237

Wind field data, 182, 233