

# **XML in Scientific Computing**

# CHAPMAN & HALL/CRC

## Numerical Analysis and Scientific Computing

### **Aims and scope:**

Scientific computing and numerical analysis provide invaluable tools for the sciences and engineering. This series aims to capture new developments and summarize state-of-the-art methods over the whole spectrum of these fields. It will include a broad range of textbooks, monographs, and handbooks. Volumes in theory, including discretisation techniques, numerical algorithms, multiscale techniques, parallel and distributed algorithms, as well as applications of these methods in multi-disciplinary fields, are welcome. The inclusion of concrete real-world examples is highly encouraged. This series is meant to appeal to students and researchers in mathematics, engineering, and computational science.

### **Editors**

Choi-Hong Lai  
*School of Computing and  
Mathematical Sciences  
University of Greenwich*

Frédéric Magoulès  
*Applied Mathematics and  
Systems Laboratory  
Ecole Centrale Paris*

### **Editorial Advisory Board**

Mark Ainsworth  
*Mathematics Department  
Strathclyde University*

Peter Jimack  
*School of Computing  
University of Leeds*

Todd Arbogast  
*Institute for Computational  
Engineering and Sciences  
The University of Texas at Austin*

Takashi Kako  
*Department of Computer Science  
The University of Electro-Communications*

Craig C. Douglas  
*Computer Science Department  
University of Kentucky*

Peter Monk  
*Department of Mathematical Sciences  
University of Delaware*

Ivan Graham  
*Department of Mathematical Sciences  
University of Bath*

Francois-Xavier Roux  
*ONERA*

Arthur E.P. Veldman  
*Institute of Mathematics and Computing Science  
University of Groningen*

Proposals for the series should be submitted to one of the series editors above or directly to:

**CRC Press, Taylor & Francis Group**  
4th, Floor, Albert House  
1-4 Singer Street  
London EC2A 4BQ  
UK

## Published Titles

### **Classical and Modern Numerical Analysis: Theory, Methods and Practice**

*Azmy S. Ackleh, Edward James Allen, Ralph Baker Kearfott, and  
Padmanabhan Seshaiyer*

### **Cloud-Computing: Data-Intensive Computing and Scheduling**

*Frédéric Magoulès, Jie Pan, and Fei Teng*

### **Computational Fluid Dynamics**

*Frédéric Magoulès*

### **A Concise Introduction to Image Processing using C++**

*Meiqing Wang and Choi-Hong Lai*

### **Decomposition Methods for Differential Equations: Theory and Applications**

*Juergen Geiser*

### **Desktop Grid Computing**

*Christophe Cérin and Gilles Fedak*

### **Discrete Variational Derivative Method: A Structure-Preserving Numerical Method for Partial Differential Equations**

*Daisuke Furihata and Takayasu Matsuo*

### **Grid Resource Management: Toward Virtual and Services Compliant Grid Computing**

*Frédéric Magoulès, Thi-Mai-Huong Nguyen, and Lei Yu*

### **Fundamentals of Grid Computing: Theory, Algorithms and Technologies**

*Frédéric Magoulès*

### **Handbook of Sinc Numerical Methods**

*Frank Stenger*

### **Introduction to Grid Computing**

*Frédéric Magoulès, Jie Pan, Kiat-An Tan, and Abhinit Kumar*

### **Iterative Splitting Methods for Differential Equations**

*Juergen Geiser*

### **Mathematical Objects in C++: Computational Tools in a Unified Object-Oriented Approach**

*Yair Shapira*

### **Numerical Linear Approximation in C**

*Nabih N. Abdelmalek and William A. Malek*

### **Numerical Techniques for Direct and Large-Eddy Simulations**

*Xi Jiang and Choi-Hong Lai*

### **Parallel Algorithms**

*Henri Casanova, Arnaud Legrand, and Yves Robert*

### **Parallel Iterative Algorithms: From Sequential to Grid Computing**

*Jacques M. Bahi, Sylvain Contassot-Vivier, and Raphael Couturier*

### **Particle Swarm Optimisation: Classical and Quantum Perspectives**

*Jun Sun, Choi-Hong Lai, and Xiao-Jun Wu*

### **XML in Scientific Computing**

*C. Pozrikidis*

# XML in Scientific Computing

C. Pozrikidis



**CRC Press**

Taylor & Francis Group

Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business

A CHAPMAN & HALL BOOK

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2013 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper  
Version Date: 20120813

International Standard Book Number: 978-1-4665-1227-6 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

**Visit the Taylor & Francis Web site at**  
<http://www.taylorandfrancis.com>

**and the CRC Press Web site at**  
<http://www.crcpress.com>

# Contents

<b>Preface</b>	<b>xi</b>
<b>Notation</b>	<b>xv</b>
<b>1 Text and data formatting</b>	<b>1</b>
1.1 Text formatting with latex and html . . . . .	2
1.1.1 Latex . . . . .	2
1.1.2 Html . . . . .	4
1.1.3 Latex compared to html . . . . .	6
1.2 Formatting with xml . . . . .	7
1.3 Usage and usefulness of xml files . . . . .	14
1.3.1 Data formatting . . . . .	14
1.3.2 Computer code formatting . . . . .	16
1.4 Constraints on structure and form . . . . .	17
1.4.1 DocBook schema . . . . .	19
1.4.2 LibreOffice Math . . . . .	20
1.5 Xml data processing . . . . .	21
1.5.1 Human processing . . . . .	21
1.5.2 Machine processing with xsl . . . . .	22
1.6 Relevance of xml in scientific computing . . . . .	25
1.6.1 Matrices . . . . .	27
1.6.2 Objects . . . . .	28
1.6.3 Data points on a graph as xml elements or C++ objects	29
1.6.4 Perl associative arrays . . . . .	33
1.6.5 Computing environments . . . . .	38
1.6.6 Summary . . . . .	38
<b>2 Xml essential grammar</b>	<b>39</b>
2.1 Xml tags . . . . .	39
2.2 Xml elements . . . . .	41
2.2.1 Element attributes . . . . .	41
2.2.2 Property listing and nesting . . . . .	42
2.2.3 Property and element tag names . . . . .	44
2.3 Comments . . . . .	45

2.4	Xml document declaration . . . . .	46
2.5	Character reference . . . . .	48
2.6	Language processing instructions . . . . .	49
2.7	Character data (CDATA) . . . . .	50
2.8	Xml root element . . . . .	52
2.9	Xml trees and nodes . . . . .	53
2.10	Document type definition and schema . . . . .	56
2.10.1	Internal document type definition (dtd) . . . . .	56
2.10.2	External document type definition (dtd) . . . . .	60
2.10.3	Xml schema definition (xsd) . . . . .	63
2.10.4	Loss of freedom . . . . .	64
2.11	<i>Xml</i> namespaces . . . . .	64
2.12	<i>Xml</i> formatting of computer language instructions . . . . .	67
<b>3</b>	<b>Xml data processing with xsl</b>	<b>71</b>
3.1	Xsl processors . . . . .	71
3.2	The main program . . . . .	73
3.2.1	Xsl code . . . . .	73
3.2.2	Root template . . . . .	74
3.2.3	Processing with the xsltproc processor . . . . .	75
3.2.4	Processing with a web browser . . . . .	75
3.2.5	Comments . . . . .	76
3.2.6	File name conventions . . . . .	77
3.2.7	Inverting namespaces . . . . .	77
3.3	for-each loops . . . . .	78
3.4	Extracting data with value-of . . . . .	81
3.5	Repeated parsing . . . . .	84
3.6	Extracting element attributes . . . . .	86
3.7	Conditional blocks . . . . .	89
3.8	Choose, when, and otherwise . . . . .	91
3.9	Variables and parameters . . . . .	92
3.10	Templates . . . . .	95
3.10.1	Matched templates . . . . .	95
3.10.2	Named templates . . . . .	101
3.10.3	Matched and named templates . . . . .	104
3.11	Splitting the code . . . . .	106
3.12	Summary of xsl elements and functions . . . . .	107
3.12.1	Core functions . . . . .	110
3.13	Passive processing and cascading stylesheets (css) . . . . .	111
<b>4</b>	<b>Computing with xml/xsl</b>	<b>113</b>
4.1	Elementary operations . . . . .	113
4.1.1	Using variables . . . . .	115
4.1.2	Internal mathematical functions . . . . .	117
4.1.3	Formatting numbers in the output . . . . .	119

4.1.4	Maximum and minimum . . . . .	120
4.1.5	Counting our blessings . . . . .	121
4.2	Templates are user-defined functions and subroutines . . . . .	121
4.2.1	Absolute value of a number . . . . .	122
4.2.2	Binary representation of a fractional number . . . . .	123
4.2.3	Binary representation of any number . . . . .	125
4.2.4	Hexadecimal representation of any number . . . . .	128
4.3	Further applications of Xslt templates . . . . .	131
4.3.1	Integral power of a number . . . . .	131
4.3.2	Highest integer with a given number of bits . . . . .	134
4.4	Square root of a number . . . . .	140
4.5	Exponential of a number . . . . .	145
4.6	Natural logarithm of a number . . . . .	148
4.7	Recursive sequences . . . . .	151
4.8	Greatest common divisor of two integers . . . . .	153
4.9	Student roster . . . . .	157
<b>5</b>	<b>Producing and importing xml data</b>	<b>164</b>
5.1	Fortran . . . . .	165
5.2	Perl . . . . .	174
5.2.1	XML::Simple . . . . .	175
5.2.2	Roots of an equation . . . . .	177
5.2.3	Real, imaginary, and complex roots . . . . .	181
5.2.4	Molecules . . . . .	182
5.2.5	Shapes . . . . .	187
5.2.6	Converting perl structures to xml data . . . . .	190
5.3	C++ . . . . .	191
5.4	Matlab® . . . . .	195
5.4.1	Generating xml data . . . . .	195
5.4.2	Using java to generate xml output . . . . .	198
5.4.3	Importing an xml document as a java object . . . . .	201
5.4.4	Arranging xml data into Matlab structures . . . . .	202
5.4.5	Navigating through an xml tree . . . . .	206
5.4.6	Summary and toolboxes . . . . .	207
<b>A</b>	<b>ASCII code</b>	<b>209</b>
<b>B</b>	<b>Perl quick reference</b>	<b>213</b>
<b>C</b>	<b>Summary of xslt elements</b>	<b>222</b>
C.1	Stylesheet declaration, import, and inclusion . . . . .	222
C.2	Alternative code . . . . .	223
C.3	Output formatting . . . . .	224
C.4	Comments, messages, and text . . . . .	226
C.5	Xml element manipulation . . . . .	227
C.6	Logical constructs . . . . .	228



C.7	Variables and parameters . . . . .	229
C.8	Templates . . . . .	230
C.9	Sorting and ranking . . . . .	232
<b>D</b>	<b>Functions called by xslt elements</b>	<b>233</b>

# Preface

*Xml* stands for extensible markup language. In fact, *xml* is not a language, but a systematic way of encoding and formatting data and statements contained in an electronic file according to a chosen tagging system. A tag may represent a general entity, a physical, mathematical, or abstract object, an instruction, or a computer language construct. The data can describe cars and trucks in a dealer's lot, the chapters of a book, the input or output of a scientific experiment or calculation, the eigenvalues of a matrix, and anything else that can be described by numbers and words.

## *Data presentation and description*

In the *xml* framework, information is described and presented in the same document, thus circumventing the need for legends and explanations. For example, we may order:

```
<breakfast> toast and eggs </breakfast>
```

Further cooking instructions can be included between the breakfast tag enclosed by the pointy brackets (<>) and its closure denoted by the slash (/).

## *Data reuse*

*Xml* data (input) can be read by a person or parsed and processed by a program (application) that produces a new set of data (output). Although the input is the same, the output depends on the interpretation of the tags formatting the data. The inherent polymorphism allows us to materialize the same original data in different ways. For example:

1. An author may write a book inserting formatting tags between words, equations, and figures according to *xml* conventions and grammar. The text (data) file can be processed to produce books with different appearances.
2. A scientist may write a finite-element code that produces output tagged according to *xml* conventions. The elements can be visualized using different graphics programs and the data can be sent to another person or program to serve as input.
3. A conversation could be transcribed using *xml* grammar and then printed on paper or sent to a telephone to be heard by the recipient. It is not necessary to duplicate the data.
4. A computer program could be written according to generic *xml* conventions. The instructions can be interpreted to produce corresponding code in a chosen programming language.

To demonstrate the concept of data sharing and reuse, we deliver the same instructions to a painter and a sculptor, and ask them to produce corresponding pieces of art. The *xml* data encapsulated in these instructions acquire meaning only when the tags describing the data are implemented by the artists to produce physical objects.

### *Scientific computing*

In scientific computing, we are accustomed to compiling and running a code (application) written in a language of our choice, such as C, C++, *fortran*, or *Matlab*<sup>®</sup>. The code utilizes parameters and input data that are either embedded in the program (monolithic structure) or read from companion input data files (modular structure). Emphasis is placed on the code and the output is generated readily by running the executable. In most applications, the code is more valuable than the output. The opposite is generally true in the *xml* framework where the data play a prominent role and may even serve to launch an application, as in the case of a telephone that rings only when it receives data.

### *Xml and scientific computing*

*Xml* has received a great deal of attention in the web programming and software engineering disciplines with reference to data encoding and storage, but far less attention in the mainstream computational science and engineering disciplines. Two main issues of interest in scientific computing are: (a) producing *xml* formatted output from code and (b) reading *xml* input from a data file, converting it into an appropriate data structure. It is revealing that computing environments familiar to scientists and engineers, such as *Matlab*<sup>®</sup>, *Mathcad*<sup>®</sup>, and *Mathematica*<sup>®</sup>, have embraced the *xml* framework and incorporated add-on libraries to facilitate the handling of *xml* input and output.

### *Goals of this book*

Currently available texts and *web* tutorials on *xml* data formatting discuss *xml* in the context of computer science with a clear focus on *web* and database programming.

The first goal of this book is to introduce and describe *xml* to scientists and engineers with some typesetting and programming experience.

The second goal is to introduce the extensible stylesheet language (*xsl*) with applications in *xml* data processing and numerical computation. Strange though it may seem, an *xsl* code is written according to *xml* conventions, that is, *xsl* is an *xml* implementation.

The third and perhaps most important goal of this book is to review possible ways of saving, importing, and sharing *xml* data in code written in programming languages used most frequently by scientists and engineers. Although references

to *latex*, *html*, *fortran 77* (simply called *fortran*), C++, and *perl* are made, only cursory familiarity with these languages is assumed and necessary explanations are given. Analogies and parallels will be drawn, and contrasts will be made with *xsl* to underline important similarities and differences in programming procedures.

This book is accompanied by a suite of computer programs and other documents arranged in directories corresponding to the book chapters and appendices.\* Internet resources and other information pertinent to *xml* are provided as links at the book website.

C. Pozrikidis

Summer, 2012

*Matlab*® is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098 USA  
Tel: 508-647-7000  
Fax: 508-647-7001  
E-mail: [info@mathworks.com](mailto:info@mathworks.com)  
Web: [www.mathworks.com](http://www.mathworks.com)

---

\*<http://dehesa.freeshell.org/XML>

# Notation

Nomenclature and font conventions adopted in the text are defined in the following table:

Symbol or word	Name or meaning
()	parentheses
[]	square brackets
{ }	curly brackets
<>	angle (pointy) brackets
->	<i>ascii</i> arrow
<i>filename</i>	name of a file
<b>sometext</b>	text typed in a file
<i>language</i>	name of a computer language
<b>line</b>	text typed in the keyboard
<b>result</b>	text shown in the screen
ENTER	Enter key in the keyboard

The names of standard computer languages are treated as regular words whose initial letter is capitalized at the beginning of a sentence and printed in lower case otherwise. File contents, typed instructions, and other data appearing on a computer screen are highlighted.