



COMPUTING FOR NUMERICAL METHODS USING VISUAL C++

Shaharuddin Salleh
Albert Y. Zomaya
Sakhinah Abu Bakar

COMPUTING FOR NUMERICAL METHODS USING VISUAL C++

Shaharuddin Salleh

Universiti Teknologi Malaysia
Skudai, Johor, Malaysia

Albert Y. Zomaya

University of Sydney
Sydney, New South Wales, Australia

Sakhinah Abu Bakar

National University of Malaysia
Bangi, Selangor, Malaysia



**WILEY-
INTERSCIENCE**

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2008 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Wiley Bicentennial Logo: Richard J. Pacifico

Library of Congress Cataloging-in-Publication Data:

Salleh Shaharuddin, 1956-

Computing for numerical methods using Visual c++ / by Shaharuddin Salleh, Albert Y. Zomaya, Sakhinah Abu Bakar.

p. cm. — (Wiley series on parallel and distributed computing)

Includes index.

ISBN 978-0-470-12795-7 (cloth)

1. Microsoft Visual C#.NET. 2. Numerical analysis—Data processing.

I. Zomaya, Albert Y. II. Bakar, Sakhinah Abu, 1982— III. Title.

QA76.73.C154S28 2007

005.2'768—dc22

2007023224

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

**COMPUTING FOR
NUMERICAL METHODS
USING VISUAL C++**



THE WILEY BICENTENNIAL—KNOWLEDGE FOR GENERATIONS

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

WILLIAM J. PESCE
PRESIDENT AND CHIEF EXECUTIVE OFFICER

PETER BOOTH WILEY
CHAIRMAN OF THE BOARD

*To our families,
for their help, courage, support, and patience.*

TRADEMARKS

Microsoft is the trademark of Microsoft Corporation Redmond, WA.

Visual Studio and Visual Studio.Net are the trademarks of Microsoft Corporation.

Visual C++ and Visual C++.Net are the trademarks of Microsoft Corporation.

Microsoft Foundation Classes is the trademark of Microsoft Corporation.

Matlab is the trademark of The Mathworks, Inc., Natick, MA.

Maple is the trademark of Waterloo Maple, Inc., Waterloo, Ontario, Canada.

Mathematica is the trademark of Wolfram Research, Inc., Champaign, IL.

PREFACE

Computing for Numerical Methods Using Visual C++ has been written to promote the use of Visual C++ in scientific computing. C++ is a beautiful language that has contributed to shaping the modern world today. The language has contributed to many device drivers in electronic equipment, as a tool in the development of many computer software programs, and as a tool for both research and teaching. Therefore, its involvement in providing the solution for numerical methods is very much expected.

Today, research has no boundary. A problem for study in a topic in research may involve people from several disciplines. A typical problem in engineering for studying the effect of chemical spills in a lake may involve engineers, chemists, biologists, medical doctors, mathematicians, economists, urban planners, and politicians. A comprehensive solution that satisfies all parties can only be produced if people from these disciplines cooperate, rather than having them acting as rivals.

Numerical computing is an important area of research in science and engineering. The topic is widely implemented in the modeling of a problem and its simulation. Additional work involves visualization, which makes the problem and its solution acceptable to the general audience. In the early days of computing in the 1960s and 1970s, the solutions to problems were mostly presented as text and numbers. A programming language like FORTRAN was the dominant tool, and there were no friendly interfaces to present the solutions. Things have improved much since then, as new advancements in hardware and software produce friendly tools based on Microsoft Windows. Numerical computing benefits much from Windows as the results from computation can now be visualized as graphs, numbers, moving images, as well as text.

We select numerical methods as the main title in the book as the concepts in this topic serve as the fundamentals in science and engineering. The importance of numerical methods has been proven as nearly all problems involving mathematical modeling and simulations in science and engineering have their roots in numerical methods. All numeric-intensive applications involving arrays and vectors have their concepts defined in numerical methods. Numerical methods discusses vital and critical techniques in implementing algorithms for providing fast, reliable, and stable solutions to these problems.

Numerical analysis is a branch of mathematics that studies the numerical solutions to problems involving nonlinear equations, systems of linear equations, interpolation and approximation for curve fittings, differentiation, integrals, ordinary differential equations, and partial differential equations. Numerical solutions are needed for these

problems as an alternative to the exact solutions, which may be difficult to obtain. The exact solutions for these problems are studied using analytical methods based on mathematical techniques and theorems. The numerical solutions produced may not be exact, but they are good enough for acceptance as they only differ by a small margin.

The scope for numerical analysis is very broad. The study involves the analytical derivation of the methods or techniques using mathematical principles and rules. The study also involves a detailed analysis of the errors between the approximated solutions and the exact solutions, so as to provide faster convergence as well as more accurate solutions.

Numerical methods is different from numerical analysis. Numerical methods is a branch of numerical analysis that specially deals with the implementation of the methods for solving the problems. The details about the derivation of algorithms and techniques for solving the problems and the analysis of errors are not in the main agenda of numerical methods. The main objective in numerical methods is applying the given methods for solving the problems. It is the implementation of the numerical methods that attracts interest from the practitioners who comprise the biggest consumer market. Engineers, scientists, and technologists belong to this group of people who view numerical methods as an important tool for solving their problems. On the other hand, numerical analysis is mostly confined to die-hard mathematicians who love further challenges in developing new numerical techniques for solving the problems.

There are several objectives in developing *Computing for Numerical Methods Using Visual C++*. First, no books on the market today discuss the visual solutions to problems in numerical methods using C++. There are similar books using software packages such as Matlab, Maple, and Mathematica. These software packages are not really primitive programming languages. They have been developed to hide the programming details and to implement the solution as a black box. In other words, software packages do not really teach the mathematical concepts and principles in solving a problem. For example, the inverse of a matrix can be computed using a single line of command in these packages. The user only needs to know the format and syntax of the command in order to produce the desired solution. It is not important for the user to know the underlying concepts in solving the problem. *Computing for Numerical Methods Using Visual C++* is one effort to integrate C++ with the visual solution to problems using numerical methods.

A student cannot be too reliant on software packages. There are cases where software packages fail to provide a solution because of the lack of special routines. For example, a software package may only support a maximum of five levels of the rectangular grids in a boundary-value problem involving partial differential equations. To produce 10 levels, the user will have to use C++ as the language because it is more flexible. Flexibility and versatility are two features in C++ that cannot be matched by any software package.

Our second objective is to promote C++ as a language for numerical computing. C++ has all the necessary ingredients for numerical computing because of its flexible language format, its object-oriented methodology, and its support for high numerical

precisions. However, in the past, the popularity of C++ has suffered from the emergence of several new languages. Among them are Java, Python, and C#. These new languages have been developed with the main objective to handle Web and network programming requirements. Other than that, C++ is still dominant and practical for implementation. Because of this reason, C++ is still popular in schools and universities, mostly for teaching and research purposes. C++ is also used widely in the manufacturing sectors such as in the design of device drivers for electronic components.

Our third objective is to make numerical problems friendly and approachable. This goal is important as the general public perception about mathematics is that it is tough, unfriendly, boring, and not applicable in daily life. A mathematician should not be placed in the basement floor of a building under the feeling that he is not important for people to meet. A mathematician can become a role model if he can exert his usefulness in a friendly and acceptable way, which can be done by making mathematics interesting and approachable through a series of friendly interfaces. A weak or average student can become motivated with mathematics if the right tools for understanding mathematics are provided.

A visual approach based on Windows in *Computing for Numerical Methods Using Visual C++* is our step in achieving this objective. The book teaches the reader on the friendly interfaces in tackling problems in numerical methods. The interfaces include buttons, dialog boxes, menus, and mouse clicks. The book also provides a very useful tool called *MyParser*, which can be used to develop various friendly numerical applications on Windows. *MyParser* is an equation parser that reads an equation input by the user in the form of a string, processes the string, and produces its solution. In promoting its use, we hide all technical details in the development of the parser so that the reader can concentrate on producing the solution to the problem. But the best part is *MyParser* is free for distribution for those who are interested.

Our last objective is to maintain links with the Microsoft family of products through the .Net platform. Microsoft is unarguably the driver in providing visual solutions based on Windows, and the .Net platform provides a common multilanguage program development for applications on Windows. As Visual C++ is one product supported in the .Net platform, there is a guarantee of continued support from Microsoft for its users. A Visual C++ follower can also enjoy the benefit of integrating her product with other products within the .Net platform with very minimum effort. This flexibility is important as migrating from one system to a different system by bringing along data and programs can be a very expensive, time-consuming, and resource-dependent affair.

In providing the solutions, this book does not provide detailed coverage of each topic in numerical methods. There are already many books on the market that do cover these topics, and we do not wish to compete against them. Instead, we focus on the development stages of each topic from the practical point of view, using Visual C++ as the tool. Knowing how to write the visual interfaces for the numerical problems will definitely contribute to guiding the reader toward the more ambitious numerical modeling and simulation projects. This objective is the main benefit that can be expected from the book. The reader can take advantage of the supplied codes to create several new projects for high-performance computing.

The book is accompanied by source codes that can be downloaded from the given Wiley website. This website will be maintained and updated by the authors from time to time. The codes have been designed to be as compact as possible to make them easy to understand. They are based on the Microsoft Foundation Classes library for providing the required user-friendliness tools. In designing the codes, we opted for the unguided (or non-wizard) approach in order to show the detailed steps for producing the output. This is necessary as the guided (or wizard) approach does not teach some key steps, and it causes the program to be extremely long in size. However, the reader should be able to convert each unguided program code to the guided code if the need arises.

In preparing the manuscript for *Computing for Numerical Methods Using Visual C++*, the authors would like to thank several people who have been directly or indirectly involved. The authors would like to thank Tan Sri Prof. Dr. Ir. Mohd. Zulkifli Tan Sri Mohd. Ghazali, Vice Chancellor of Universiti Teknologi Malaysia, for his forward vision in leading the university towards becoming a world-class university by 2010. Special thanks also to Professor Dr. Alias Mohd. Yusof and Professor Dr. Md. Nor Musa from Universiti Teknologi Malaysia, and Professor Stephan Olariu from Old Dominion University, for their support and encouragement.

Shaharuddin Salleh
Albert Y. Zomaya
Sakhinah Abu Bakar
April 2007

CODES FOR DOWNLOAD

All codes discussed in this book can be downloaded from the following URL:

ftp://ftp.wiley.com/public/sci_tech_med/computing_numerical

The files in the URL are organized into a directory called SALLCode. The codes for the program are located in the folders bearing the chapter numbers; for example, Code4 is the project for Chapter 4. The files included are the executable (.exe), header (.h), C++ (.cpp) and the parser object file, MyParser.obj.

The ftp site will constantly be maintained and updated. Any questions, comments, and suggestions should be addressed to the first author at ss@utm.my.

The system requirements for the codes are as follows:

Intel Pentium-based Personal computer with 256 MB RM and above.

Microsoft Windows 1998 and above.

Microsoft Visual C++ version 6, and above.

All code files have been tested using Microsoft Visual C++.Net version 2003. The same files should be compatible with Visual C++ version 6 and below and with Microsoft Visual C++.Net version 2005 and above.

CONTENTS

Preface	xiii
Codes for Download	xvii
1. Modeling and Simulation	1
1.1 Numerical Approximation	1
1.2 C++ for Numerical Modeling	3
1.3 Mathematical Modeling	4
1.4 Simulation and Its Visualization	6
1.5 Numerical Methods	7
1.6 Numerical Applications	7
2. Fundamental Tools for Mathematical Computing	13
2.1 C++ for High-Performance Computing	13
2.2 Dynamic Memory Allocation	14
2.3 Matrix Reduction Problems	22
2.4 Matrix Algebra	35
2.5 Algebra of Complex Numbers	43
2.6 Number Sorting	51
2.7 Summary	54
Programming Challenges	55
3. Numerical Interface Designs	56
3.1 Microsoft Foundation Classes	56
3.2 Graphics Device Interface	57
3.3 Writing a Basic Windows Program	60
3.4 Displaying Text and Graphics	68
3.5 Events and Methods	69
3.6 Standard Control Resources	71
3.7 Menu and File I/O	78
3.8 Keyboard Control	87
3.9 MFC Compatibility with .Net	92
3.10 Summary	95

4. Curve Visualization	96
4.1 Tools for Visualization	96
4.2 MyParser	96
4.3 Drawing Curves	106
4.4 Generating Curves Using MyParser	115
4.5 Summary	126
Programming Challenges	126
5. Systems of Linear Equations	127
5.1 Introduction	127
5.2 Existence of Solutions	128
5.3 Gaussian Elimination Techniques	131
5.4 LU Factorization Methods	142
5.5 Iterative Techniques	161
5.6 Visualizing the Solution: Code5	172
5.7 Summary	189
Numerical Exercises	190
Programming Challenges	192
6. Nonlinear Equations	193
6.1 Introduction	193
6.2 Existence of Solutions	194
6.3 Bisection Method	195
6.4 False Position Method	198
6.5 Newton–Raphson Method	201
6.6 Secant Method	203
6.7 Fixed-Point Iteration Method	206
6.8 Visual Solution: Code6	208
6.9 Summary	225
Numerical Exercises	225
Programming Challenges	226
7. Interpolation and Approximation	227
7.1 Curve Fitting	227
7.2 Lagrange Interpolation	228
7.3 Newton Interpolations	231
7.4 Cubic Spline	239
7.5 Least-Squares Approximation	244
7.6 Visual Solution: Code7	249
7.7 Summary	264
Numerical Exercises	265
Programming Challenges	265

8. Differentiation and Integration	267
8.1 Introduction	267
8.2 Numerical Differentiation	268
8.3 Numerical Integration	271
8.4 Visual Solution: Code8	279
8.5 Summary	286
Numerical Exercises	286
Programming Challenges	287
9. Eigenvalues and Eigenvectors	288
9.1 Eigenvalues and Their Significance	288
9.2 Exact Solution and Its Existence	289
9.3 Power Method	291
9.4 Shifted Power Method	292
9.5 QR Method	294
9.6 Visual Solution: Code9	302
9.7 Summary	322
Numerical Exercises	322
Programming Challenges	323
10. Ordinary Differential Equations	324
10.1 Introduction	324
10.2 Initial-Value Problem for First-Order ODE	325
10.3 Taylor Series Method	327
10.4 Runge–Kutta of Order 2 Method	330
10.5 Runge–Kutta of Order 4 Method	333
10.6 Predictor–Corrector Multistep Method	335
10.7 System of First-Order ODEs	338
10.8 Second-Order ODE	341
10.9 Initial-Value Problem for Second-Order ODE	342
10.10 Finite-Difference Method for Second-Order ODE	345
10.11 Differentiated Boundary Conditions	351
10.12 Visual Solution: Code10	358
10.13 Summary	378
Numerical Exercises	378
Programming Challenges	380
11. Partial Differential Equations	381
11.1 Introduction	381
11.2 Poisson Equation	385
11.3 Laplace Equation	394
11.4 Heat Equation	397

xii CONTENTS

11.5	Wave Equation	406
11.6	Visual Solution: Code 11	411
11.7	Summary	437
	Numerical Exercises	437
	Programming Exercises	438
Index		441

Modeling and Simulation

- 1.1 Numerical Approximations
- 1.2 C++ for Numerical Modeling
- 1.3 Mathematical Modeling
- 1.4 Simulation and Its Visualization
- 1.5 Numerical Methods
- 1.6 Numerical Applications
- References

1.1 NUMERICAL APPROXIMATIONS

Numerical methods is an area of study in mathematics that discusses the solutions to various mathematical problems involving differential equations, curve fittings, integrals, eigenvalues, and root findings through approximations rather than exact solutions. This discussion is necessary because the exact solutions to these problems are difficult to obtain through the analytical approach. For example, it may be wise to evaluate

$$\int_{-1}^5 x \sin x \, dx,$$

as the exact solution can be obtained through a well-known technique in calculus called *integration by parts*. However, it is not possible to apply the same method or any other analytical method to solve

$$\int_0^2 \frac{e^{-x}}{3 \sin x + x^2} \, dx.$$

The given equation in the above integral is difficult to solve as it is not subject to the exact methods discussed in ordinary calculus. Therefore, a numerical method is needed to produce a reasonably good approximated solution. A good approximated solution, whose value may differ from the exact solution by some fractions, is definitely better than nothing.