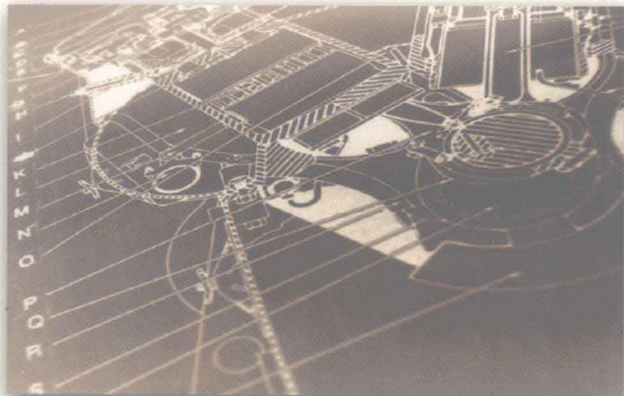
BASIC ENGINEERING SERIES AND TOOLS

# MATHCAD

## A TOOL FOR ENGINEERING PROBLEM SOLVING



PHILIP J. PRITCHARD

# Mathcad

# A Tool for Engineering Problem Solving

**Philip J. Pritchard**
***Manhattan College***



WCB McGraw-Hill

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Lisbon London Madrid
Mexico City Milan New Delhi Seoul Singapore Sydney Taipei Toronto

*WCB/McGraw-Hill*

*A Division of The* ***McGraw-Hill*** *Companies*



MATHCAD: A TOOL FOR ENGINEERING PROBLEM SOLVING



This book is printed on acid-free paper.

6 7 8 9 0 DOC/DOC 0 9 8 7 6 5 4

ISBN 0–07–012189–3

Vice president and editorial director: *Kevin T. Kane*
Executive editor: *Eric Munson*
Developmental editor: *Holly Stark*
Marketing manager: *John T. Wannemacher*
Project manager: *Alisa Watson*
Production supervisor: *Lori Koetters*
Compositor: *Shepherd, Inc.*
Typeface: *10/12 Century Schoolbook*
Printer: *R. R. Donnelly & Sons Company*

**Library of Congress Cataloging-in-Publication Data**

Pritchard, Philip J.
Mathcad : a tool for engineering problem solving / Philip Pritchard.
p. cm.
Includes index.
ISBN 0–07–012189–3
1. Engineering—Data processing. 2. MathCAD. I. Title.
TA345.P765 1999
620'.00285'5369—dc21 97–47318

# Mathcad

# A Tool for Engineering Problem Solving

## MCGRAW-HILL'S BEST

*A Beginner's Guide to Technical Communication,* by A. Eisenberg, 0–07–092045–1
*C Program Design for Engineers,* by A. Tan and T. D.'Orazio, 0–07–913678–8
*Introduction to Engineering Design,* by A. Eide, et al., 0–07–018922–6
*Introduction to Engineering Problem Solving,* by A. Eide, et al., 0–07–021983–4
*Introduction to Fortran 90/95,* by S. Chapman, 0–07–011969–4
*Introduction to MATLAB® for Engineers,* by W. Palm, 0–07–047328–5
*Mathcad: A Tool for Engineering Problem Solving,* by P. Pritchard, 0–07–012189–3
*Mathsoft's Student Edition of Mathcad 7.0,* 0–07–292600–7
*Spreadsheet Tools for Engineers: Excel 97 Version,* by B. Gottfried, 0–07–024654–8

# Preface

This book was written as part of the McGraw-Hill *BEST* series (Basic Engineering Series and Tools). The intended audience for all of the books in this series is the introductory engineering class. However, as I was writing the book, it became apparent that, due to the nature of Mathcad itself, even a text covering only its main features would end up covering some material to which a beginning engineering student might not yet have been introduced. Hence, although the primary market for this text is still the introductory engineering class, in truth engineering students from freshmen through seniors, and even graduate students, will find it helpful in learning Mathcad. Practicing engineers who want a compact guide to using Mathcad for solving their engineering problems will probably find it useful as well.

Many colleges in the United States are now using a one- or two-semester course designed to introduce students to the basics of what it means to be an engineer. These courses invariably include exposing the students to use of the personal computer as a communication and analysis tool. For example, they may be taught how to use a spreadsheet, a technical calculation package such as Mathcad, and how to use a programming language such as *C* for doing engineering work. This book is suitable for use in such a course.

Each chapter introduces features of Mathcad by immediately doing engineering examples, so that the student can see that the features being described do have a real-world, practical engineering application. The chapters should be read while at the computer, so that the student can replicate each of the examples. In doing so, they will not only learn how to use Mathcad, but also get exposure to some typical engineering problem-solving methodologies.

Some of the chapters have exercises at the end of each chapter section. These are intended as practice exercises on the specific material covered in that section. It's probably a good idea that the student do all of these. Exercises at the end of each chapter are intended for further practice, and would make good homework questions.

Sophomores, Juniors, and Seniors will also find the book useful in learning Mathcad. Depending on their previous experience with Mathcad, they should be able to move fairly quickly through practically all of the material, except for perhaps one or two sections (for example, Laplace transforms, covered in Chapter 9). These students will include people who have never used Mathcad and those who have some knowledge of it. The former will be able to self-teach by reading the book from the beginning. The latter will be able to pick and choose those topics they wish to learn more about. In fact, the Mathcad 7 Student Edition has a number of important changes from the previous version, not least of which are the equation-editing techniques and the additional built-in symbolic features such as Laplace transforms.

Graduate students and engineers in industry should have no difficulty in reading this book from cover to cover (while using the computer) in a relatively brief period of time. After doing so, they will have a good grounding in the basic features of Mathcad and a sense of its power in solving the engineering problems they encounter.

What is Mathcad, anyway? We'll discuss this in detail in Chapter 1, but here let's just say it's one of several technical calculation applications that are available today. Although it is always difficult to make such predictions, it seems likely that in the next few years it may become the dominant application of this kind. In my opinion, if it does so, it will be well deserved. It has done especially well in its migration from the DOS world to the graphic world of Windows (in its various incarnations of Windows 3.x, Windows 95, etc.). You have probably experienced software that is very "clunky," that requires you to work with several windows within the application, that is just plain ugly, or that has a very odd menu structure or icon system. That software will probably be one that existed before Windows and has never completely migrated to the fully graphic interface. This description certainly does not apply to Mathcad because it has, again in my opinion, more successfully taken to the Windows environment than any other mathematics application. It has a completely graphic interface and is truly WYSIWYG (what-you-see-is-what-you-get).

This is one of the reasons it is my most-used and best-liked computer application (with the runner-up spot going to the spreadsheet). There are lots of other reasons. For example, it is especially good at handling units. My experience with engineering students (including graduate students) over the years has taught me that even the best of them have a somewhat casual attitude to units. (Question: "What's the acceleration of gravity?" Answer: "32.2"). Mathcad will automatically work with the units for you and give you the answer in any units you wish. It might appear that this is a bad thing, similar to giving a calculator to someone who is innumerate, but actually it's not. In my experience with teaching Mathcad, students get a better understanding for units. This is because, if a student tries to do something using units that is improper, Mathcad will "flag" that error, forcing the student to check their own use of units. For example, if a student inadvertently defines a mass in, say *lbf* rather than *lb,* sooner or later Mathcad will compute something that will have units that the student will recognize to be incorrect. In this way Mathcad will reinforce correct units usage, while at the same time eliminating all of the drudgery of unit conversions (from, for example, a density expressed in $lb/ft^3$ to one in $kg/m^3$).

To their credit, the people at Mathsoft have continued to develop Mathcad at a healthy pace. They seem to release a new version about once every twelve to eighteen months, and each new version genuinely has major improvements. For example, the much-improved equation editing of version seven is a major change. There also seems to be a pattern: as each new version comes along, it seems that what was only in the more expensive professional edition in a previous version pretty much gets inserted into the new inexpensive student edition, and a lot of new features are put in both editions. This is true with the Mathcad 7 Student Edition: it has most of the capability of Mathcad 6 Professional Edition (for instance it now has symbolic transforms such as Laplace, Fourier and Z), and Mathcad 7 Professional Edition has new extensive programming capabilities. Mathcad is also kept very current. For example, in the last few years the explosion in attention given to the World Wide Web is reflected in the fact that version seven has built-in web connectivity (and in fact, so did version six). Hence, users of Mathcad have one more avenue for remaining up to date on developments in the world of computing.

I'd like to thank a number of people for their help as I prepared this book. Professors Dan Haines, Bahman Litkouhi, Mohammad Naraghi, and Graham Walker, my students Steve Rutgerson and Suzanne Wright, all of Manhattan College, and Professor Byron Gottfried of the University of Pittsburgh offered many suggestions for improvements. Edward Adams of Adams Technologies gave me many good ideas for exercise questions. The people at Mathsoft, especially Paul Lorczak and Clay Stone, have been very helpful, and are to be thanked for producing a marvelous piece of software. Finally, David Shapiro (who initially contacted me about writing such a book) and especially Eric Munson and Holly Stark (both of whom have a remarkable ability to be simultaneously informal and professional), and all of McGraw-Hill, have earned my respect and thanks for their support. Of course, any praise this book earns should be directed at them, and any errors are mine and mine alone.

# Dedication

To my mother, Elaine, for all her tea over the years, and my wife, Penelope, for her sympathy during the struggle to complete this book on schedule.

# About the Author

Philip J. Pritchard received his Ph.D. in engineering mechanics from Columbia University in 1987. He has been a faculty member of mechanical engineering at Manhattan College since 1981, where he teaches undergraduate and graduate courses in thermodynamics, fluid mechanics, and analysis and numerical methods, and has twice been the recipient of the teacher of the year award from the Manhattan College student section of Pi Tau Sigma. He publishes and presents papers in the area of engineering education, specifically in the use of the PC (personal computer) for doing engineering analysis. Dr. Pritchard is a member of Sigma Xi and the American Society for Engineering Education.

# Contents

1

# What Is Mathcad and Why Use It?

What tools do engineers and scientists need to do their job? To answer this question, we need to consider what it is that engineers and scientists do.

Engineers and scientists typically apply the concepts and laws of engineering science and physics to physical phenomena and, based on these, develop mathematical models to simulate (or represent) the phenomena. These models can take many mathematical forms: at their simplest, the model results in an algebraic equation, but sometimes the analysis results in a differential equation or an integral equation. Very often an analysis can lead to not one equation but a set of equations.

As an illustration of this, let's consider an example from electrical engineering. Suppose you have the DC circuit shown in Figure 1.1. *If you're not familiar with DC circuits, don't worry, because the focus of this text is not engineering theory but how to use Mathcad to solve the engineering mathematics problems that result from doing engineering analyses.*

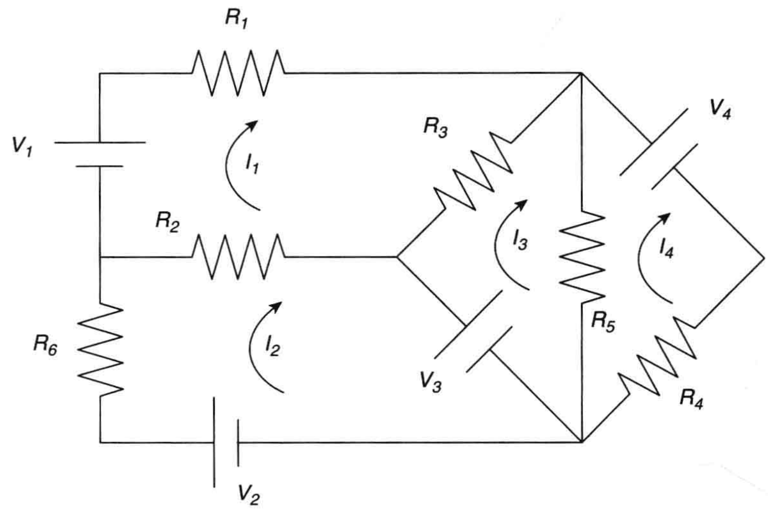



**Figure 1.1**

Suppose the resistors $R_1$ through $R_6$ and applied voltages $V_1$, $V_2$, $V_3$, and $V_4$ are all known. This fairly complicated circuit must be analyzed to find the currents $I_1$, $I_2$, $I_3$, and $I_4$ in each loop. In Figure 1.1 we have *assumed* that all four currents are clockwise, so if we find after solving the problem that one of them is negative, it will mean that that particular current runs counterclockwise. Note also that the current in a particular part of the circuit can be deduced from these currents. For example, the current through $R_3$ will be $I_1 - I_3$.

How do we solve such a problem? To solve it, the engineer uses the appropriate concepts or ideas from electrical engineering to *mathematically* model the circuit. In this case, we can use the idea that the total voltage change around a closed loop is zero. Here we have four unknowns, so we will need four equations. It can be shown that these equations are:

$$\begin{aligned} V_1 - I_1(R_1 + R_2 + R_3) + I_2R_2 + I_3R_3 &= 0 \\ V_2 - I_2(R_2 + R_6) + I_1R_2 - V_3 &= 0 \\ V_3 - I_3(R_3 + R_5) + I_1R_3 + I_4R_5 &= 0 \\ -V_4 - I_4(R_4 + R_5) + I_3R_5 &= 0 \end{aligned} \tag{1.1}$$

How would you go about solving these equations? Well, you could manipulate the equations to try and eliminate all the unknowns except, say, $I_1$, and then continue on to find the other three unknowns. However, this would be extremely tedious to do, and also it would be easy to make an arithmetical error at some point. Imagine if you had a more complex circuit, with, for example, 100 current loops. You'd end up with 100 equations for these 100 unknowns. We obviously need a better method than manipulating these equations by hand. The approach used is called *linear algebra* (and we'll see how to use Mathcad to do this in Chapter 6). To implement this approach, we will need *mathematical tools*.

This example demonstrates, and is typical of, the situation engineers and scientists almost always face: after using engineering or physics concepts to develop a mathematical model of a phenomenon, mathematical tools are needed to get a solution to the problem. Today these tools invariably involve using a computer.

On the other hand, engineers sometimes need to deal not with an *equation* or *set of equations* but with *experimental data*, which they need to mathematically manipulate in some way to extract useful information. An example of this might be calibration data for a pressure gage:

| Reading | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_{act}$ (Pa) | 0 | 5 | 10 | 15 | 20 | 30 | 40 | 50 | 60 | 70 |
| $p_{ind}$ (Pa) | 0 | 6 | 11 | 13 | 24 | 32 | 38 | 56 | 61 | 73 |