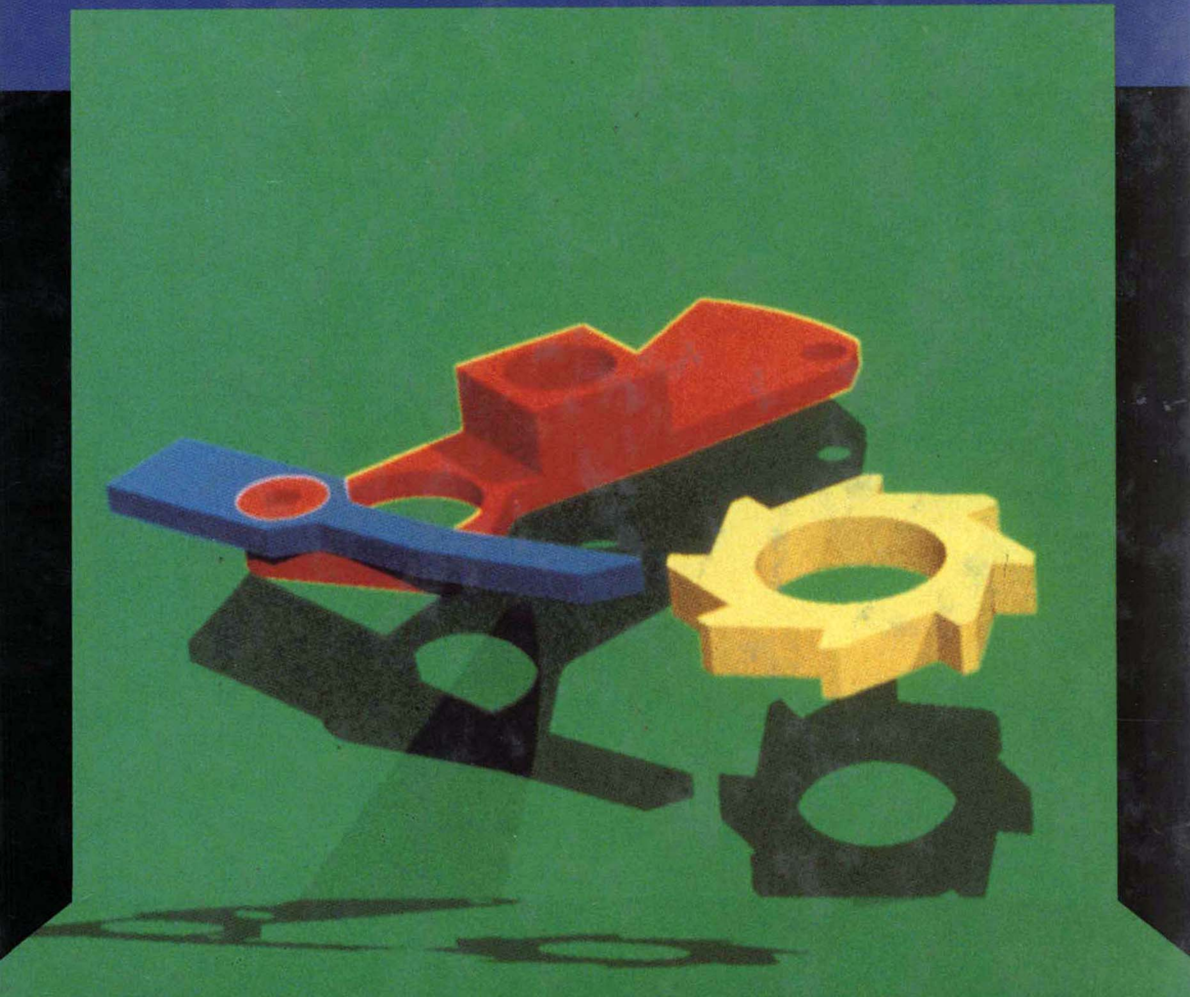
Computer-Aided Design



DEAN L. TAYLOR

This book is in the Addison-Wesley series in **Mechanical Engineering**

Cover: Photo courtesy of Jon Davis, General Electric Company.

Library of Congress Cataloging-in-Publication Data

Taylor, Dean, 1949-
Computer aided design / Dean Taylor.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-16891-X
1. Computer-aided design. I. Title.
TA174.T39 1992
620'.0042'0285—dc20 91-23895
CIP

1 2 3 4 5 6 7 8 9 10-MA-95 94 93 92 91

To Kathy and Laurie

Color Plate Insert Credit List

Plate 1. Surface plot for pressure distribution within fluid film of crank end of a connecting rod, with zero pressure shaded red. Courtesy of Gary LaBouff.

Plate 2. Iteration map for Newton's method applied to Project 3.1. Courtesy of Scott A. Burns, University of Illinois at Urbana-Champaign.

Plate 3. Patch Representation of joystick handle. Courtesy of Silicon Graphics, Inc.

Plate 4. Shaded surface of joystick handle. Courtesy of Silicon Graphics, Inc.

Plate 5. Two-dimensional color map.

Plate 6. Pressure distribution over surface of transport aircraft. Courtesy of Douglas Aircraft Company, McDonnell Douglas Corporation.

Plate 7. Color shaded surface stress distribution for human femur. Courtesy of Cornell-HSS Program in Biomechanical Engineering.

Plate 8. Contoured surface stress distribution for a mechanical part. Courtesy of PDA Engineering.

Plate 9. Surface distribution of von Mises stress. Courtesy of PDA Engineering.

Plate 10. Temperature distribution during molding of plastic part. Courtesy of Cornell Injection Molding Project.

Plate 11. A three-dimensional rendering of a mechanical part. The colors show different levels of stress for complex loading conditions calculated with PATRAN P/Stress, a stress analysis program. Surface rendering allows the stresses on the surface of the part to be visualized. Created by Pixar. Data courtesy of Mr. Harris Hunt, PDA Engineering. ©1987 Pixar. All rights reserved.

Plate 12. A three-dimensional rendering of a mechanical part. The colors show different levels of stress for complex loading conditions calculated with PATRAN P/Stress, a stress analysis program. Volume rendering allows the internal stresses within the part to be visible. Created by Pixar. Data courtesy of Mr. Harris Hunt, PDA Engineering, ©1987 Pixar. All rights reserved.

Plate 13. Surface representations of isocontours for volume filling data. Low pressure (white), low streamwise velocity (yellow), outward ejections of low-speed fluid (red), and wallward sweeps of high-speed fluid (lavender).

Plate 14. Three-dimensional instantaneous velocity vectors, with blue representing low pressure and magenta representing high pressure.

Plate 15. Injection molding sequence, $T = T_4$. Courtesy of Richard Ellson, Kodak Research Scientist; Donna Cox, Visualization Research Artist; Ray Idaszak, Visualization Research Programmer; National Center for Supercomputer Applications.

Plate 16. Two dimensional velocity vector plot using marker arrows. Courtesy of PDA Engineering.

Plate 17. Three dimensional velocity vector plot using market arrows. Courtesy of PDA Engineering.

Plate 18. Pressure distribution on fan with flow ribbons. Courtesy of Douglas Aircraft Company, McDonnell Douglas Corporation.

PREFACE

This book is the result of a course (MAE 389, Computer-Aided Design) that the author has taught for 12 of the past 15 years in the Mechanical and Aerospace Engineering School at Cornell University. Computer-aided design has been evolving rapidly as a field, with changes in hardware and software tools available. However, a core of material has developed that will serve the student in future years as hardware and software continue their rapid change. Unfortunately, that core of material has been available only by using three or four separate books, an unsatisfactory solution for an undergraduate course. The objective of this book is to present the fundamentals of computer-aided design, showing its analytical rigor and algorithmic nature independent of specific software implementation.

The objective of the book is not to teach any particular CAD program. Focusing on a specific program would be counter to the trends in this field and would also limit the scope of material. In order to use existing software effectively and create usable macros or programs, students must understand the computing environment and the underlying algorithms. The objective is to produce knowledgeable CAD users who can quickly learn a specific program within a specific environment and who, understanding the fundamental concepts and limitations of techniques and data structures, can utilize that program to the "limits of the envelope."

The author's basic philosophy is programming oriented and project oriented. Exercises have been included in the book to help guide readers who may be using the book outside of a formal classroom setting. The projects in the book are taken from real situations and use realistic values as much as possible. One subgoal of this book is to provide a set of realistic interconnected design problems. The projects have been chosen so that the course can satisfy ABET design course objectives with the addition of appropriate reports and supplementary material.

Project courses are particularly challenging when combined with a substantial amount of formal material. A single large project is attractive because of the scope it makes possible, but a series of individual projects is also attractive because of the more uniform workload for the students. The course at Cornell has been based on a series of related projects, as reflected in the book.

A course could be organized in several ways from this book. If the only prerequisite is a programming course, then the CAD course might include Chapters 2, 3, 5, 6, and 7. However, if the students have already taken a numerical methods course (in addition to a programming course), the CAD course could be based on Chapters 4, 5, 6, 7, 9, and 10. If the existing curriculum includes courses in finite-element analysis, simulation or optimization, these chapters can be replaced with material from Chapters 8, 10, and 11. The author has also taught a second course using the more advanced material in Chapters 7, 8, 9, 10, and 11.

The prerequisite for the course at Cornell is the College of Engineering Introductory Computing course (taught in PASCAL on Apple Macintosh). However, the majority of the students have also taken an introductory numerical methods course (taught in FORTRAN). The CAD course uses material from Chapters 4, 5, 6, 7, and 9. Students are assigned a very simple initial project (usually from Chapter 2) to familiarize themselves with the computing environment (the operating system, editor, and compiler, and the input/output and file system operations). This is followed by the sailboat projects from Chapters 5, 6, 7, and 9. By the end of the semester, the students have produced a working simulation of a sailboat, performance predictions, and a graphical representation of the hull. The final project is to tie all of this together and participate in a class regatta. A communications package is provided so that each student's program can be advised of the other students' boat locations. The situation is similar to that of multiple-machine, multiple-aircraft flight simulators. Although the regatta is clearly a competition, the students' grades are determined by participation rather than by order of finish.

Commercial programs are available for all the material covered in this book (although no single program spans the entire range). Although a course could be organized using three or four commercial packages, the course offered at Cornell is oriented to programming, and students are expected to create programs to complete the design tasks. Obviously, one does not create such programs from scratch, so a collection of software should be provided. This is similar to the situation in industry, where some software will be available but must be adapted to the immediate task. Depending upon the material covered and projects assigned, the class should be provided with a complete numerical methods library (including curve-fitting, splines, polynomial root-finding, matrix decomposition, variable time step integration, and unconstrained and constrained optimization), a hierarchical graphics package at the level of PHIGS or GL, a geometry package, an interactive graphing and plotting program, a contour plotting and/or surface plotting program, and a menu creation program. The author has been able to obtain or create public domain alternatives for many of these. Students are also encouraged to make use of programs on personal computers (generally Macintosh at Cornell) for plotting, spreadsheet and prototyping, and document preparation.

This book, and the course that has used it, assumes that the students have previous experience in programming, understand an operating system and file system, and can learn programming productivity tools such as a full screen editor and debugger. The students are therefore expected to pick up the particular environment for this course with moderate effort. Specifically, the course uses the UNIX operating system. The course could be taught in many alternative languages. A procedural compiled language is most appropriate. The author has recently chosen to teach in C, although FORTRAN is certainly a viable alternative. Given the project orientation of the book, and the excellent interlanguage procedural calling capability, a course could be offered in both languages, with each student making an individual choice.

The course at Cornell uses Silicon Graphics UNIX workstations, and students prepare projects in C. With some compromise in graphics capability, almost any UNIX workstation could be used. For the course as offered at Cornell, current personal computers are not sufficiently powerful. The primary difficulty is the lack of virtual memory, which imposes constraints on program and data size. This is a problem if students are to use efficiently a collection of programs and software that is not designed to work together.

I am indebted to the following reviewers who helped at various times throughout the book's development: Thomas R. Chase, University of Minnesota; Mark S. Ganter, University of Washington; Mark Henderson, Arizona State University; David A. Hoeltzel, Columbia University; K. Khozeimeh, George Washington University; Gary L. Kinzel, Ohio State University; Arvid Myklebust, Virginia Polytechnic Institute; Thomas Phelps, University of Missouri—Columbia; Donald R. Riley,, University of Minnesota; and Robert R. Ryan, University of Michigan.

I want to thank my colleagues at Cornell—Don Bartel, Jack Booker, an Paul Dawson—who have provided stimulation and support. Much of my approach to CAD developed from discussions with graduate students over many years: Randy Bartlett, David Bell, Michael Butler, Shirish Chinchalkar, Av Edidin, Steve Fontes, Paul Hauck, Kumar Hebbale, Pat Hollis, Bill Horn, Diana Hauser, Nobuhisa Kamikawa, B. R. Kumar, Govirdaraj Kuntimad, Xuehai Li, Mike McCarthy, Masa Moriyama, Dev Prabhu, Ganesh Subbarayan, Scott Walter, DaZong Wang, and Richard Warkentin. As do all university faculty, I owe a tremendous debt to the students I have been able to work with. I would like to express my appreciation to the teaching assistants and students of MAE 389 and 489 during the past 15 years.

Pat Hollis provided much of the material for Chapters 3 and 4. Paul Dawson provided projects for Chapter 4. Jack Booker provided the material for the annotated bibliography for Chapters 4 and 5. Ganesh Subbarayan provided material for Chapter 6. Paul Hauck provided the material for the annoted bibliography for Chapters 7 and 8.

I want to express appreciation to Gary Weigand, who taught me about graphics programming, and to acknowledge the staff of CADIF, which has provided the computing environment for this class in recent years. I also want to thank Mike Piplani and John Lottey. I would also like to acknowledge IBM, which has supported the IMAP research lab for many years.

I would like to thank Ginny Giles, who helped prepare the manuscript, and Teresa Howley, who helped prepare the illustrations. My thanks to my editor, Don Fowley, and associate editor Laurie McGurie, whose gentle prods saw the project to completion. Thanks also to Kristina Williamson and Melissa G. Madsen, who provided an extensive and valuable edit of the entire manuscript.

This book was prepared in TEX, and the majority of the figures were prepared in Adobe Illustrator.

D. L. T.

CONTENTS