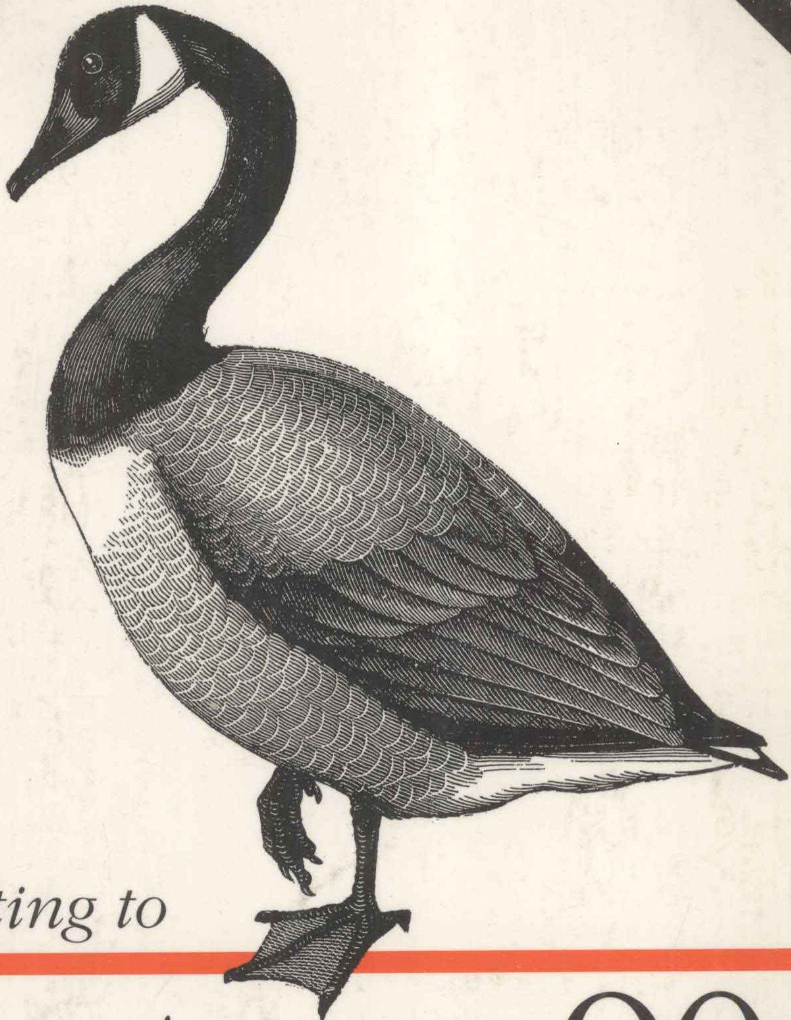
Programming Languages

Lay-flat Binding



Migrating to

Fortran 90

A NUTSHELL HANDBOOK

James F. Kerrigan

O'Reilly & Associates, Inc.

Migrating to Fortran 90

Jim Kerrigan

O'Reilly & Associates, Inc.
103 Morris Street, Suite A
Sebastopol, CA 95472

Migrating to Fortran 90
by Jim Kerrigan

Editor: Andy Oram
Production Editor: Leslie Chalmers

Printing History:

November 1993: First Edition.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly & Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

This book is printed on acid-free paper with 50% recycled content, 10-15% post-consumer waste. O'Reilly & Associates is committed to using paper with the highest recycled content available consistent with high quality.

ISBN: 1-56592-049-X

Dedicated to the memory of
William M. Kerrigan

Preface

A huge volume of scientific and engineering code is developed and maintained by an extensive community of Fortran programmers. Since the last ANSI standard was delivered fifteen years ago, sixty books specifically targeting the "Fortran 77" readership remain in print, attesting to the vigor of the language. Since microprocessors first became a viable computing platform, well over a dozen commercial Fortran compilers for PCs have come to market, proving a demand for the language.

Fortran 90 has many features not found in the previous standard. Individually, these features are sufficiently "new" to require full and detailed description. Collectively, they extend the functionality of the language and broaden its applicability.

This book is tailored to the current Fortran practitioner. It includes dozens of full program examples. Fortran programs have always balanced the need of implementing some algorithm against an independent need to acquire information, communicate with the user, and generate reports and output files. Both the formula translation need and the data management need are addressed by the examples in this book. Algorithms are drawn from the natural sciences, mathematics, and operations research. Data management and program control are illustrated by the demands of the interactive prompt/response cycle, data structures, linked lists, and input/output error handling.

Reader experience encompasses not only Fortran 77 but many extensions that have spread throughout the industry; this book recognizes that. Comparisons to popular extensions and work-arounds as well as Fortran 77 can be found here. Furthermore, perfect knowledge of standard Fortran 77 is not assumed; recapitulations are included for obscure features.

Finally, this book lists additional Fortran 90 resources. It includes pointers to available full and partial Fortran 90 compilers, addresses from which to order the actual Fortran 90 standard, and detailed instructions on how to get machine-readable copies of the source code for the examples in this book.

Who Should Read This Book?

This book is for programmers who are already working with large applications in some dialect of Fortran. This community knows Fortran essentially as implemented by compilers subscribing to Fortran 77, the ANSI standard published in 1978. Fortran 90 is the next step in the evolution of the language.

This book is a comprehensive explanation of every new feature that the 1990 ANSI/ISO standard has added to the language. The text describes every change from the Fortran 77 standard, and grounds new concepts in the readers' current needs and experiences.

How This Book Is Organized

This book is structured to deliver the information in the order that busy programmers would most likely want it. Simple enhancements and widely useful features come early in the book. Features that require a lot of time and program redesign to implement, or that are less useful to the majority of programmers, come later. But every change between Fortran 77 and Fortran 90 is covered somewhere in the book.

Following the above criteria, Chapter 1, *Compatibility*, offers a brief look at the whole language and the effect of migration on current Fortran programmers. Chapters 2 through 7 offer the most substantial new features, while Chapters 8 through 10 offer miscellaneous enhancements that are useful in particular situations. Chapter 11 covers any new intrinsic functions that were not fully described in earlier chapters.

Arrays are the essential repositories of data in Fortran, and their use has been widely enhanced with a number of powerful extensions over the years. These extensions, now standardized, are fairly easy to learn and useful in almost every programming situation. They are the subject of Chapter 2, *Array Operations.*

Derived types—also known as records in VAX FORTRAN and structures in C—complement arrays well. They are not as widely useful as the new array features, but have many applications in grouping different types of data. They are covered in Chapter 3, *Derived Types.*

Chapters 4, 5, and 6 form a progression, taking the reader through features of increasing complexity and power. The overall subject is program organization, involving the grouping of data and subprograms. Chapter 4, *Subroutines and Functions Revisited,* starts the process by examining the common model for the subdivision of programs. It offers some new ways of handling subprograms with minimal disruption of current practice. Along with program structure, it is useful to look at the option of invoking user-defined functions and subroutines through operators, so that feature comes next in Chapter 5, *Overloaded Operators.* Finally, Chapter 6, *Modules,* offers a full solution to the problems addressed in Chapter 4, promoting the use of modules for new programs and for major changes to old ones.

Chapter 7, *Dynamic Memory Management,* discusses the last major feature introduced by Fortran 90, the dynamic management of memory through allocatable arrays and pointers. While not central to program structure like the features of the previous chapters, the features in this chapter have important uses.

Chapter 8, *File Handling,* is rather disjointed. Here, all the new features for file handling and I/O are presented in one place. Fortran 90's changes in these areas are small but in some cases long-desired, such as the ability to specify whether a file is opened at the beginning or the end.

Numeric precision has been hard to specify in a portable manner up to now. The new features of Fortran 90 standardizing this area appear in Chapter 9, *Numeric Models.*

The special programming topic of bit manipulation is covered in Chapter 10, *Bit Functions.*

New intrinsic functions (75 of them) intertwine with the other new programming capabilities offered by Fortran 90. Most of the new intrinsics, because they interact heavily with other features, appear in the first 10 chapters of the book. Chapter 11, *Intrinsic Functions,* serves as a summary and clean-up, covering the less important intrinsics or the subtler features that did not appear earlier.

Finally, several appendices list full Fortran 90 compilers, partial Fortran 90 compilers, ordering procedures for the standard itself, and the complete source code of some of the larger example programs.

How to Improve This Book

Fortran 90 is very new. There is very little practical experience with the full language anywhere. Some of the lack of experience is due to the availability of compilers. Several compilers have been delivered to the marketplace, one available as early as mid-summer 1991, and more scheduled for delivery in late 1993 and early 1994. But they have been used primarily in an experimental role. Only now are they being gradually adopted for production use. Hardware and software vendors will probably migrate existing compilers towards Fortran 90 in direct response to their users' discovery of how new Fortran 90 features assist in research and development.

Some of the lack of experience is due to the limited exposure Fortran 90 has had in the press. This book is one of few devoted entirely to Fortran 90. Others are:

Aberti, Christophe

1992 Fortran 90: Initiation à partir du FORTRAN 77. Série Informatique Éditions. 144 pages. Menton, France.
ISBN 2-090615-00-6

Adams, Jeanne C., et. al.

1990 Fortran 90 Handbook: Complete ANSI/ISO Reference. McGraw-Hill Publishing Co. 740 pages. New York, NY.
ISBN 0-070004-06-4

Brainerd, Walter S. et. al.

1990 A Programmer's Guide to Fortran 90. McGraw-Hill Publishing Co. 410 pages. New York, NY.
ISBN 0-070002-48-7

1993 (German edition) R. P. Oldenbourg Verlag gmbh. Munich, Germany.

Counihan, Martin

1991 Fortran 90. Pitman Publishing. 309 pages. London, England.
ISBN 0-273030-73-6

Delannoy, C.

1993 Programmer en Fortran 90: Guide Complet. Edition Eyrolles. 413 pages. Paris, France.
ISBN 2-212087-23-2

Gehrke, Wilhelm

1991 Fortran 90 Referenz-Handbuch: der neue Fortran-Standard. Carl Hanser Verlag. 964 pages. München, Germany.
ISBN 3-446163-21-2

Heisterkamp, Manfred

1991 Fortran 90: eine informelle Einführung. BI-Wissenshaftsverlag. 202 pages. Mannheim, Germany.
ISBN 3-411153-21-0

Langer, Erasmus

1993 Programmieren in Fortran 90. Springer-Verlag. 320 pages. New York, NY.
ISBN 0-387824-46-4

Lignelet, Patrice

1993 Fortran 90: Approche par la Pratique. Série Informatique Éditions. 240 pages. Menton, France.
ISBN 2-090615-01-4

Meissner, Loren P.

1994 Fortran 90. PWS-KENT Publishing Company. 650+ pages. Boston, MA.
ISBN 0-534933-72-6

Metcalf, Michael and John K. Reid

1990 Fortran 90 Explained. Oxford University Press. 294 pages. Oxford, England.
ISBN 0-198537-72-7

1993 (French edition) AFNOR. Paris, France.

1994 (Russian edition) Mir. Moscow, Russia.

1994 (Japanese edition) Kyoritsu Shuppan. Tokyo, Japan.

Morgan, J. Steve and J. Lawrie Schonfelder

1993 Programming in Fortran 90. Blackwell Scientific Publications. 360 pages. Oxford, England.
ISBN 0-632028-38-6

Überhuber, C. and P. Meditz

1993 Software-Entwicklung in Fortran 90. Springer-Verlag. 426 pages. Berlin, Germany.
ISBN 0-387824-50-2

Wojcieszynski, Ranier and Brigitte Wojcieszynski

1993 Fortran 90 Programmieren mit dem Neuen Standard. Addison-Wesley Publishing Co., Inc. 344 pages. Reading, MA.
ISBN 3-893196-00-5

Four college-level textbooks cover both Fortran 77 and Fortran 90 in an integrated way:

Bronson, Gary J.

1992 Modern Fortran 77/90. Scott/Jones Inc. 595z pages. El Granada, CA.
ISBN 0-962423-05-X.

Ellis, T. M. R.

1990 Fortran 77 Programming with an Introduction to the Fortran 90 Standard. Addison-Wesley Publishing Company. 641 pages. Wokingham, England.
ISBN 0-201416-38-7.

Koffman, Elliot B. and Frank L. Friedman

1993 Fortran with Engineering Applications. Addison-Wesley Publishing Company. 664 pages. Reading, MA.
ISBN 0-201558-75-0.

Zirkel, Gene and Eli Berlinger

1993 Understanding Fortran 77 and 90. PWS-KENT Publishing Company. 608 pages. Boston, MA.
ISBN 0-534934-47-1

In journals, Fortran 90 has received significant coverage only in the Fortran Journal published by the Fortran User Group (Fullerton, CA) and in Fortran Forum published by the Association for Computing Machinery (New York, NY). Fortran 90 reached the "popular" press only in short articles in BYTE (Doris Appleby, "FORTRAN", Volume 16, Number 9, pages 147-150, September 1991 issue), the New Scientist (Michael Metcalf, September 12, 1992 issue, pages 30-33), and Scientific American (Paul Wallich, "FORTRAN Forever", Volume 265, Number 1, page 112, July 1991 issue).

Your experience is invited. New examples, better examples, clearer examples are urgently sought. If you discern an application of Fortran 90 features that isn't even hinted at in this book, speak up. As you discover shortcomings in the text, note them and send them in. Mistakes? Some remain regardless of the care that went into rooting them out: please let the publisher know so that they can be corrected in a future edition.

Further Developments on the Fortran Front

It is still not clear what kinds of problems are best solved with Fortran 90. Certainly Fortran 90 will inherit a whole universe of applications from Fortran 77. Fortran 90 might gain ground for Fortran by providing a viable alternative for problems that would have been tackled in C or C++.

Extensions to Fortran 90 for parallel programming are the objective of the High Performance Fortran Forum organized at the Center for Research on Parallel Computation at Rice University (Houston, TX). Their "High Performance Fortran" (HPF) facilitates writing data parallel programs where the distribution of data impacts performance. It supports high performance programming on a wide variety of machines, including massively parallel SIMD and MIMD systems and vector processors. The HPF Forum is on target to complete the specification of this extended Fortran by late winter, 1993. In fact, a subset of this high performance Fortran called xHPF will be available commercially in early spring, 1993, as a joint product of Applied Parallel Research (Placerville, CA) and Kuck and Associates, Inc. (Champaign, IL).

System level Fortran 90 bindings are planned as the POSIX 1003.19 standard by the Institute of Electrical and Electronics Engineers, Inc. (Piscataway, NJ). This effort was scheduled to complete in 1996. Basically, the committe was charged with a mission to recast the Fortan 77 language interface (POSIX 1003.9) to system services (POSIX 1003.1) into the "best practices" of Fortran 90. Work began in 1991 but stopped in July 1993. Further development has been postponed because vendors cited incomplete experience in development of Fortran 90 compilers and users cited a lack of widespread experience in using the new Fortran 90 constructs. As such, the project to develop POSIX 1003.19 has been withdrawn, but is likely to be reintroduced as broad-based industry Fortran 90 experience grows.

Examples Online

The source code and input files for all examples in this book are available free from UUnet (that is, free except for UUnet's usual connect-time charges). If you have access to uunet, you can retrieve the examples using UUCP or FTP. For UUCP, find a machine with direct access to uunet, and type the following command:

```
uucp uunet\!~/nutshell/fortran90/fortran90.tar.Z yourhost\!~/yourname/
```

The backslashes can be omitted if you use the Bourne shell (sh) instead of the C shell (csh). The file should appear some time later (up to a day or more) in the directory **/usr/spool/uucppublic/yourname**.

You don't need to subscribe to UUNET to be able to use their archives via uucp. By calling 1-900-468-7727 and using the login "uucp" with no password, anyone may uucp any of the UUNET's on line source collection. (Start by copying uunet!~/ls-lR.Z, which is a compressed index of every file in the archives.) As of this writing, the cost is 40 cents per minute. The charges will appear on your next telephone bill.

To use ftp, you will need to find a host with direct access to the Internet. A sample session is shown, with commands in boldface:

```
% ftp uunet.uu.net
Connected to uunet.uu.net.
220 uunet FTP server (Version 5.99 Wed May 23 14:40:19 EDT 1993) ready.
Name (uunet.uu.net:yourname): anonymous
331 Guest login ok, send ident as password.
Password: yourname@yourhost.xxx
230 Guest login, ok, access restrictions apply.
ftp> cd nutshell/fortran90
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> get fortran90.tar.Z
200 PORT command successful.
150 Opening ASCII mode data connection for fortran90.tar.Z
```

```
226 Transfer complete.
ftp> quit
221 Goodbye.
%
```

The file is a compressed tar archive. To extract the files once you have retrieved the archive, type:

```
% zcat fortran90.tar.Z | tar xf -
```

System V systems require the following tar command instead:

```
% zcat fortran90.tar.Z | tar xof -
```

All of the examples have been run on a Sun 3/50 system under SunOS 4.1 with Version 1.1(392) of the f90 compiler from the Numerical Algorithms Group Ltd. If you notice behavior on your system that show the examples to be in error, please contact the author through the publisher at nutshell@ora.com.

Acknowledgments

I am grateful to many people for their help during the course of this book's development.

Tony Nilles of the Numerical Algorithms Group championed my request within NAG for an extended loan of their Fortran 90 compiler. In addition, the assistance of Karl Knapp and Malcolm Cohen of NAG's technical support staff was prompt and comprehensive whenever I posed a question.

Bob Bonocore and Rich Bateman of Sun Microsystem's local office came to the rescue with much needed advice when my aged, but serviceable, Sun 3/50 system threatened to give up the ghost.

Dick Arscott (Software Associates), Walt Brainerd (Unicomp), John McCalpin (University of Delaware), Loren Meissner (University of San Francisco), and Mike Metcalf (CERN), all generously gave their time to review the completed manuscript. It has profited immensely from their knowledge and experience.

Mike Loukides and Steve Talbott at O'Reilly & Associates took a chance on my proposal and approved the book's development. I thank them for accepting the risk of publishing a book by an unknown author on a subject that in and of itself contained a large number of unknowns.

Edie Freedman, Leslie Chalmers, and Chris Reilley at O'Reilly & Associates handled all the design, production, and illustrations, respectively. Although

I am responsible for what this book contains, they deserve the credit for how good it looks. Linda Walsh and Jane Appleyard put immense energy into marketing the book among vendors of Fortran 90 compilers.

Andy Oram at O'Reilly & Associates edited this book. That is a brief statement of his role, but falls far short of Andy's influence. Andy is an expert in organizing written material to convey complex ideas: the flow of topics in this book is testament to his skill. Furthermore, his knowledge of the subject kept the treatment on an even keel. Finally, throughout the process, Andy made every effort to produce the best book possible, and, like a director, he suggested, cajoled, requested, and demanded the same from me. Andy's influence is the foundation of whatever measure of quality this book achieves.

My family invested so much in this book. My boys—Ian, Phillip, and Evan—watched me work all hours on this project and never complained. My wife, Ellen, missed my company as I became more and more involved, but she understood my commitment. This book would never exist were it not for my family's support.

Table of Contents