Fortran Resources [1]

Ian D Chivers Jane Sleightholme

September 21, 2022

[1]The original basis for this document was Mike Metcalf's Fortran Information File. The next input came from people on comp-fortran-90. Details of how to subscribe or browse this list can be found in this document. If you have any corrections, additions, suggestions etc to make please contact us and we will endeavor to include your comments in later versions. Thanks to all the people who have contributed.

Revision history

The most recent version can be found at
`https://www.fortranplus.co.uk/fortran-information/`
and the files section of the comp-fortran-90 list.
`https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=comp-fortran-90`

September 2022. Updates to the compiler chapter and parallel programming chapter.

May 2021. Major update to the Intel entry. Also changes to the editors and IDE section, the graphics section, and the parallel programming section.

October 2020. Added an entry for Nvidia to the compiler section. Nvidia has integrated the PGI compiler suite into their NVIDIA HPC SDK product. Nvidia are also contributing to the LLVM Flang project. Updated the 'Additional Compiler Information' entry in the compiler section. The Polyhedron benchmarks discuss automatic parallelisation. The fortranplus entry covers the diagnostic capability of the Cray, gfortran, Intel, Nag, Oracle and Nvidia compilers. Updated one entry and removed three others from the software tools section. Added 'Fortran Discourse' to the e-lists section. We have also made changes to the Latex style sheet.

September 2020. Added a computer arithmetic and IEEE formats section.

June 2020. Updated the compiler entry with details of standard conformance.

April 2020. Updated the Fortran Forum entry. Damian Rouson has taken over as editor.

April 2020. Added an entry for Hewlett Packard Enterprise in the compilers section

April 2020. Updated the compiler section to change the status of the Oracle compiler.

April 2020. Added an entry in the links section to the ACM publication Fortran Forum.

March 2020. Updated the Lorenzo entry in the history section.

December 2019. Updated the compiler section to add details of the latest release (7.0) of the Nag compiler, which now supports coarrays and submodules. Updated the compiler section to have details of the latest Intel release. See the compiler section for more details. Added a section on compiler comparisons to the compiler chapter. Major update to the parallel section to bring up to date with current offerings. Added entries in the parallel section of how to configure Ubuntu to support coarray and MPI programming. Added an entry in the parallel section on how to configure cygwin to support MPI programming. Updated and renamed the Numeric Libraries chapter to Numeric Libraries and software repositories.

October 2019. Added details of a new book by Subrata Ray, called Fortran 2018 and Parallel Programming. Also added a book entry to the history section.

September 2019. Added the Numeric Libraries chapter

September 2019. Spell checked the document using the TeXnicCenter spelling tool. Updated the Fortran 2003 compiler conformance table with information from Arm and NEC.

August 2019. Updated the history section to have the publication dates of all ISO Fortran standards. Updated the Fortran 2003 compiler conformance table. Added ISBNs to several books and added initials to several of the authors. Updated the compilers section.

June 2019. Added ISBNs to several of the books. Added links to the ACM, IEEE and British Library in the on line resources chapter.

March 2019. Updated the history chapter.

January 2019. Split the chapter that had Fortran historic information and other material into a Fortran history chapter and other links chapter. Added additional historic material.

December 2018. Added 2 new books that were published in 2018, and 3 new books due to be published in 2019.

November 2018. Updated the Fortran 2008 compiler standard conformance data.

October 2018. Added an entry for Flang. Added details of PGI Community Edition under free compilers. Added details of Flang under free compilers. Updated the Oracle entry. Corrected compiler conformance details for some of the compilers. Moved Pathscale to no longer available. Corrected an entry in the book section. Added details of a new book. Added details of a new course from Sorcery Inc, Writing Fortran 2018 Today: Object-Oriented Parallel Programming. Made typesetting changes due to changes with Adobe Acrobat pdf format.

April 2018, added an entry in the compiler section for SimplyFortran. Updated the compiler section to have summary details of standard conformance. Added a new book and added 2018 to the book title section. Major update to the J3 and WG5 entries in the standard section to bring up to date with recent changes.

April 2018, updated the Nag entry to have details of the 6.2 release. Add a compiler entry for Arm. Added 2 new books. Updated the plusFORT entry.

March 2017, updated the Oracle entry to have details of the 12.6 beta.

December 2016, updated the gfortran entry to add details about using the Windows subsystem for Linux option. Thanks to Paul Richard Thomas for the post to comp.lang.fortran about this.

June 2016, updated the Pathscale compiler entry and added details of their Dogfood IDE in the IDE section

May 2016, added entries for Nag Fortran Builder for the Windows and Apple Mac operating systems.

May 2016, added a new section on Object Oriented Programming, in Other Web Links, with link to a site with very good coverage of the subject.

May 2016, added some entries to the Fortran History section, in Other Web Links.

May 2016, Minor corrections; added Oracle 12.5 beta

April 2016, Added PLplot and GTK+ entries

April 2016, Updated the Nag and Oracle compiler entries. Added a new book. Added a new course under the NAG entry.

July 2015. Updated the Cray compiler entry - Thanks to Bill Long for providing up to date information. Changed various entries in the course section. Most notable were adding an entry for Sourcery Inc and the Archer service at Edinburgh, and deleting the Hector entry. Added Doctran (cross-platform documentation generation tool for the Fortran) to the Tools section.

June 2015. Added coverage of free compilers, compilers free for personal use, and Intel's free compiler offer for educational users. Added Chivers and Sleightholme third edition, due 14 August 2015

10 March 2015. Corrected missing German ß in an address. Thanks to Rolf Maier for pointing this out.

9 March 2015. Added an entry for coarray.org in the Coarray section of the parallel programming entry. Editorial changes: page size; long urls split over two lines; minor setting corrections.

30 January 2015. Added an entry for Microsoft Visual Studio 2013 Community Edition. This free version is equivalent to Microsoft Visual Studio 2013 Professional. Updated the Nag entry in the Tools section to provide details of the capability of the Nag compiler. The compiler can now produce call graphs, do a dependency analysis, produce interface modules and pretty print or polish Fortran code.

3 September 2014, Added entries for two ides, Code::Blocks and geany. The information came from an interchange on linkedin.

10 February 2014, Added an errata entry for the The Fortran 2003 Handbook and an index entry for the same book. Thanks to Dick Hendrickson and John Harper for posting to comp-fortran-90 about these items.

February 2014, Checked various entries and web links. Missed some urls with leading and trailing spaces in an earlier update, and these have now been fixed. Corrected some spelling.

November 29 2013. Added one new book entry and updated an existing book entry.

August 6 2013 Minor changes to the IDE chapter. Also corrected urls that have leading spaces after the opening brace.

Contents

Chapter 1

Books

Version 1.22, October 2019; Added an entry for a new book from Subrata Ray, called Fortran 2018 and Parallel Programming.

Version 1.21, August 2019; Added ISBN's to several books and corrected several entries using the British Library catalogue and Abebooks.

Version 1.20, June 2019; Added ISBN's to several books and corrected several entries using the British Library catalogue.

Version 1.19, March 2019; Added a section on the early history of Fortran.

Version 1.18, December 2018; Added details of several new books, łposted to comp.lang.fortran by beliavsky - thanks for the heads up on these books.

version 1.17, October 2018; Added details of Chivers and Sleightholme fourth edition and corrected Chivers and Sleightholme, third edition entry.

version 1.16, April 2018; Added details of the new MRC book.

Version 1.15, April 2018; Added the Curcic book; Updated the Chapman entry.

Version 1.14, April 2016; Added the latest Walt Brainerd book.

Version 1.13, June 2015; Added the latest Chivers and Sleightholme book.

Version 1.12, 10 February 2014; Updated the Fortran 2003 Handbook entry.

Version 1.11, February 2014; Corrected some spelling.

Version 1.10, November 2013;Added an entry for the Hanson and Hopkins book. Also updated the entry on Adams, Brainerd et al, which is now available from Springer as an ebook in PDF format.

Version 1.9, May 2013; An errata list for Modern Fortran Explained, Metcalf, Reid and Cohen is now available.

Version 1.8, January 2013; Added Section 1.1, Metcalf, Reid, Cohen; Thanks to Anton Shterenlikht for pointing out this omission!

Version 1.7, August 2012; Added Section 1.1, Chivers; Added Section 1.1, Markus;

Version 1.6, January 2012;Added Section 1.5 Chivers;

Version 1.5, October 2011;Added Section 1.1 Chivers; Added Section 1.1 Clerman;Added Section 1.1 Gnu Fortran; Added Section 1.1 Rouson;Added Section 1.2 Lakshmivarahan;

Version 1.4, July 2010;Added Section 1.1 Brainerd; Added Section 1.1 McCormmack;Added Section 1.1 Ray; Added Section 1.2 Rajaram;Added Section 1.3 Barlow et al; Added Section 1.4 Chandra et al;Added Section 1.4 Chapman et al;

Version 1.3, June 2009;Removed invalid web address in Morgan and Schonfelder entry; Corrected spelling in Chinese entry;

Version 1.2, September 2008;Added Section 1.1, Adams et al; Added Section 1.10, Ciaburro;

Version 1.1, September 2007;Added Section 1.1, Chapman; Added Section 1.2, Lemmon;

1.1 Fortran 2003, 2008 and 2018 - English

Adams, J.C., Brainerd, W.S., Hendrickson, R.A., Maine, R.E., Martin, J.T., Smith, B.T., The Fortran 2003 Handbook, The Complete Syntax, Features and Procedures, 2008, Springer Verlag, ISBN: 978-1-84628-378-9. This book is also available as an ebook in PDF format. An errata file is available at

`http://www.fortran.com/F03HB_errata.html`

and an additional set of index entries at

`http://homepages.ecs.vuw.ac.nz/~harper/f2003handbookindex`

Ekrem Aydiner and Orhan Gemikonakli, Computational Methods in Science and Engineering : Models, Algorithms, Coding, and Analysis with Gnu FORTRAN, Chapman and Hall, July 2019, ISBN-13: 9781498798419 ISBN-10: 1498798411

Brainerd, W.S., Guide to Fortran 2008 Programming, 2015, Springer Verlag, ISBN 978-1-4471-6758-7

Brainerd, W.S., Guide to Fortran 2003 Programming, 2009, Springer Verlag, ISBN 978-1-84882-542-0

Chapman S.J., Fortran For Scientists and Engineers, 4th edition, 2018, McGraw-Hill.ISBN-10: 0073385891 ISBN-13: 978-0073385891

Chapman S.J., Fortran 95/2003 For Scientists and Engineers, 2007, McGraw-Hill. ISBN 978-0073191577, ISBN 0073191574

Chivers I.D., Sleightholme J., Introduction to Programming with Fortran, Springer Verlag. ISBN 978-3-319-75501-4 (Hardback), 978-3-319-75502-1 (eBook) Fourth Edition, 2018

Chivers I.D., Sleightholme J., Introduction to Programming with Fortran: With coverage of Fortran 90, 95, 2003, 2008 and 77. Springer Verlag. ISBN 978-3-319-17700-7, Third Edition, 2015

Chivers I.D., Sleightholme J., Introduction to Programming with Fortran: With coverage of Fortran 90, 95, 2003, 2008 and 77. 2012, Springer Verlag. ISBN-10: 0857292323 ISBN-13: 978-0857292322, Second Edition

Chivers I.D., Sleightholme J., Introduction to Programming with Fortran: With coverage of Fortran 90, 95, 2003 and 77. 2006, Springer Verlag. ISBN 1-84628-053-2, First Edition

Clerman N.S., Spector W., Modern Fortran: Style and Usage Cambridge University Press, 2011. ISBN-10: 052173052X ISBN-13: 978-0521730525

Curcic Milan, Modern Fortran - Building Efficient Parallel Applications. Manning Publications. Began March 2018 Publication in Early 2019 (estimated) ISBN 9781617295287 475 pages (estimated) printed in black and white.

`https://www.manning.com/`

Fehr H., Kindermann F., Introduction to Computational Economics Using Fortran, OUP, 2018, ISBN: 9780198804406

Gnu Fortran Team, Using Gnu Fortran: Manual For Gcc Version 4.3.3, CreateSpace. ISBN-10: 1441412662 ISBN-13: 978-1441412669, 2009

Richard J. Hanson, Tim Hopkins, Numerical Computing with Modern Fortran. Publisher: Society for Industrial & Applied (November 30, 2013) Language: English ISBN-10: 1611973112 ISBN-13: 978-1611973112

Izaac J., Wang J., Computational Quantum Mechanics, Springer, February 2019, ISBN 978-3-319-99930-2

Markus A., Modern Fortran in Practice, 2012, Cambridge University Press, ISBN-10: 1107017904 ISBN-13: 978-1107017900

McCormack D., Scientific Software Development with Fortran, 2009, ISBN 978-1-4452-5445-6

Metcalf M., Reid J. and Cohen M., Modern Fortran Explained, second edition, Oxford University Press. 2018. ISBN 9780198811893 (hardback); ISBN 9780198811886 (paperback)

Metcalf M., Reid J. and Cohen M., Modern Fortran Explained, 2011, Oxford University Press. ISBN 978-0-19-960141-7, ISBN 978-0-19-960142-4. Errata list is available at

`ftp://ftp.numerical.rl.ac.uk/pub/MRandC/edits.pdf`

Metcalf M., Reid J. and Cohen M., Fortran 95/2003 Explained, 2004, Oxford University Press. ISBN 0-19-852693-8, ISBN 0-19-852692-X

Ray Subrata, Fortran 2018 with Parallel Programming, Chapman and Hall and CRC Press, Taylor Francis Group, August 2019, ISBN 9780367218430

Ray S., A Textbook on Fortran 2003, 2009, Alpha Science International, ISBN 978-1-84265-479-8

Rouson D., Xia J., Xu X., Scientific Software Design: The Object-Oriented Way Cambridge University Press, 2011. ISBN-10: 0521888131 ISBN-13: 978-0521888134

Rouson D., Filippone S., Shende S., Modern Fortran: Tricks of the Software Engineering Trade, Taylor and Francis, July 2019, ISBN 10: 1138051918, ISBN 13: 9781138051911

Trapp M., Ochsner A., Computational Plasticity for Finite Elements: A Fortran-Based Introduction, Springer, 2018, ISBN 978-3-319-77206-6

1.2 Fortran 95 - English

Adams J.C., Brainerd W.S., Martin J.T,, Smith B.T., and Wagener J.L, Fortran 95 Handbook, 1997, MIT. ISBN 0-262-51096-0.

Adams J.C., Brainerd W.S., Martin J.T.and Smith B.T., Fortran Top 95, Ninety Five Key Features of Fortran 95, 10 The book is only available in PDF form from the Fortran Store,

`http://www.fortran.com/`

Akin E., Object Oriented Programming via Fortran 90/95, 2003, Cambridge University Press. ISBN 0-521-52408-3.

Chapman S. J., Fortran 90/95 for Scientists and Engineers, 2004, McGraw Hill. ISBN 0-07-282575-8.

Chapman S.J., Introduction to Fortran 90/95, 1997 McGraw-Hill, 1997, ISBN 0-07-011969-4.

Chivers I.D., Sleightholme J., Introducing Fortran 95, 2000, Springer Verlag. ISBN 185233276X

Counihan M., Fortran 95, 1997, UCL. ISBN 185728367-8.

Etzel M., Dickinson K., Digital Visual Fortran 90 Programmer's Guide, 1999, Digital Press. ISBN 1-55558-218-4.

Gehrke W., Fortran 95 Language Guide, 1996, Springer ISBN 3-540-76062-8.

Lakshmivarahan S., Sudarshan K. Dhall, Programming in Fortran 90/95, Publisher: Pearson Custom Publishing (2002) ASIN: B000XM3WZ0

Lawrence N., Compaq Visual Fortran: A Guide to Creating Windows Applications, 2001, Digital Press. ISBN 1-55558-249-4.

Lemmon D.R.,Schafer J.L., Developing Statistical Software in Fortran 95, 2005, Springer. ISBN-10 0387238174 ISBN-13 978-0387238173.

Metcalf M., Reid J., Fortran 90/95 Explained, 1999 Oxford University Press. ISBN 0-19-850558-2

Morgan J.S.& Schonfelder J.L., Programming in Fortran 90/95, available in Acrobat PDF format from

`http://www.fortran.com/`

Rajaram V., Computer Programming in Fortran 90 and 95, 2004, Prentice Hall, ISBN 978-8120311817

Vowels R., Introduction to Fortran 90/95, Algorithms, and Structured Programming, Part 1: Introduction to Fortran 90, Part 2: Algorithms and Fortran 90. ISBN 0-9596384-8-2.

1.3 Fortran 90 - English

Barlow R.J., Barnett A.R., Computing for Scientists: Principles of Programming with Fortran 90 and C++, 1998, Wiley Blackwell, ISBN 978-0471955962

Adams J.C., Brainerd W.S., Martin J.T., Smith B.T.and Wagener J.L., Fortran 90 Handbook, 1992, McGraw Hill. ISBN 0-07-000406-4.

Brainerd W.S., Goldberg C.H., and Adams J.C., Programmer's Guide to Fortran 90, 1996, Springer ISBN 0-387-94570-9.

Brainerd, Goldberg and Adams., Programmer's Guide to Fortran 90, 1994, Unicomp. ISBN 0-07-000248-7

Brooks D.R., Problem solving with Fortran 90: for scientists and engineers, 1997, Springer. ISBN 0-387-98229-9.

Chivers I.D., Sleightholme J., Introducing Fortran 90, 1995, Springer. ISBN 3-540-19940-3

Ellis T.M.R., Philips I.R., Lahey T.M., Fortran 90 Programming, 1994, Addison Wesley, ISBN 0-201-54446-6.

Etter D.M., Fortran 90 for Engineers, 1995, Benjamin/Cummings ISBN 0-201544-46-6.

Gehrke, Fortran 90 Language Guide, 1996, Springer ISBN 3-540-19926-8

Hahn D.B., Fortran 90 for Scientists and Engineers, 1994, Edward Arnold ISBN 0-340-60034-9.

Huddleston J.V., Fortran 90, 1996, Exchange Publ. Div., ISBN 0-945261-07-1.

Kerrigan J.F., Migrating to Fortran 90, 1993, O'Reilly. ISBN 1-56592-049-X

Meissner L.P., Fortran 90, 1995, PWS Kent ISBN 0-534-93372-6.

Metcalf M., Reid J., Fortran 90 Explained, 1990 Oxford University Press, ISBN 0-19-853772-7

Morgan J.S.& Schonfelder J.L., Programming in Fortran 90, 1993, Alfred Waller. ISBN 1-872474-06-3

Nyhoff L., Leestma S., Fortran 90 for Engineers and Scientists1, 1996, Prentice Hall, ISBN 0-13-519729-5.

Nyhoff L., Leestma S., An introduction to Fortran 90 for Engineers and Scientists, 1996, Prentice Hall, ISBN 0-13-505215-7.

Redwine C., Upgrading to Fortran 90, 1995, Springer-Verlag, ISBN 0-387-97995-6.

Schick W., Silverman G., Fortran 90 and Engineering Computation, 1994, John Wiley, ISBN 0-471-58512-2.

Smith I.M., Programming in Fortran 90, 1995, Wiley ISBN 0471-94185-9.

Wagener J.L., Fortran 90 Concise Reference, 1998, Absoft. ISBN 0-9670066-0-0.

1.4 English books on related topics

Advanced Scientific Computing - Wille, Wiley, 1995, ISBN 0471-95383-0.

Atlas for Computing Mathematical Functions ... in Fortran 90 and Mathematica - Thompson, Wiley, 1997, 0-471-18171-4.

Contemporary Computing for Technical Engineers and Scientists: using Fortran 90 and spreadsheets - Forsythe, PWS, 1997, 0-534-93139-1.

Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing, Volume 2 of Fortran Numerical Recipes - Press, Teukolsky, Vetterling and Flannery, Cambridge U. Press, ISBN 0-521-57439-0, 1996. Code can be downloaded (purchased) from

`http://www.nr.com/`

A CDROM is also available (see Web site).

1.5 Chinese

Introduction to Programming with Fortran, with coverage of Fortran 90, 95, 2003 and 77. Ian Chivers and Jane Sleightholme. Posts and Telecom Press and Springer-Verlag London. 2009. ISBN 978-7-115-21227-6. (A translation of Introduction to Programming in Fortran).

Programming Language Fortran 90 - He Xingui, Xu Zuyuan, Wu Qingbao and Chen Ming yuan, China Railway Publishing House, Beijing, ISBN 7-113-01788-6/TP.187, 1994.

Fortran 90 - Walter S. Brainerd, Charles H. Goldberg, Jeanne C. Adams, CHEP, Beijing, and Springer, Berlin, 2000, ISBN 7-04-007937-2 (a translation of Programmer's Guide to Fortran 90).

1.6 Dutch

Fortran 90 - W.S. Brainerd, Ch.H. Goldberg, and J.C. Adams, translated by J.M. den Haan, Academic Service, 1991, ISBN 90 6233 722 8.

1.7 Finnish

Fortran 90/95 - Juha Haataja, Jussi Rahola and Juha Ruokolainen. Center for Scientific Computing (Finland), 2001, 339 pages, 3rd edition, ISBN 952-9821-60-3. WWW version:

`http://www.csc.fi/oppaat/f95/`

1.8 French

Fortran 90; Approche par la Pratique - P. Lignelet, Serie Informatique Editions, Menton, 1993, ISBN 2-909615-01-4.

Fortran 90. Les concepts fondamentaux, the translation of Fortran 90 Explained, M. Metcalf, J. Reid, translated by M. Caillat and B. Pichon, AFNOR, 1993, Paris, ISBN 2-12-486513-7.

Fortran 90; Initiation a partir du Fortran 77 - Aberti, Serie Informatique Editions, Menton, 1992, ISBN 2-909615-00-6.

Les specificites du Fortran 90, M. Dubesset et J. Vignes, Editions Technip, 1993 . ISBN 2-7108-0652-5.

Manuel complet du langage Fortran 90, et guide d'application, P. Lignelet, Serie Informatique Editions, 1995. ISBN 2-909615-02-2.

Manuel Complet du Langage Fortran 90 et Fortran 95, Calcul intensif et Genie Logiciel, P. Lignelet, Masson Editions, Paris, 1996, ISBN 2-225-85229-4.

Programmer en Fortran 90, C. Delannoy, Eyrolles, 1992, ISBN 2-212-08723-3.

Traitement des donnees numeriques avec Fortran 90, M. Olagnon, Masson Editions, 1996, ISBN 2-225-85259-6.

Structures des donnees (et leurs algorithmes) en Fortran 90 et 95, P. Lignelet, Masson Editions, Paris, ISBN 2-225-85373-8.

1.9 German

Fortran 90 Lehrbuch - D. Rabenstein, Hanser, 1995, ISBN 3-446-18235-7.

Die Programmiersprache FW. Gehrke, Springer-Verlag, ISBN 3-540-63376-6.

Fortran 90 - B.Wojcieszynski and R.Wojcieszynski, Addison-Wesley, 1993, ISBN 3-89319-600-5.

Fortran 90: eine informelle Einfuhrung - M. Heisterkamp, BI-Wissenschaftsverlag, 1991, ISBN 3-411-15321-0.

Fortran 90 Kurs: technisch orientiert - G. Schmitt, Oldenbourg, 1996, ISBN 3-486-23896-5

Fortran 90, Lehrund Arbeitsbuch fur das erfolgreiche Programmieren, W.S. Brainerd, C.H. Goldberg, and J.C. Adams, translated by Peter Thomas and Klaus G. Paul, R. Olbenbourg Verlag, Muenchen, 1994, ISBN 3-486-22102-7.

Fortran 90 Lehr- und Handbuch - T. Michel, BI-Wissenschaftsverlag, 1994.

Fortran 90 Referenz-Handbuch: der neue Fortran-Standard - W. Gehrke, Carl Hansen Verlag, 1991, ISBN3-446-16321-2.

Programmierung in Fortran 90 - Schobert, Oldenburg, 1991.

Programmierung mit Fortran 90 - Baumer, Vieweg, Braunschweig, 1997, ISBN 3-528-05208-2.

Programmieren in Fortran - Erasmus Langer, Springer-Verlag, Wien New York, 1993. ISBN 3-211-82446-4, 0-387-82446-4.

Software Entwicklung in Fortran 90 - Uberhuber and Meditz, Springer Verlag, 1993, ISBN 0-387-82450-2.

1.10 Italian

Titolo: Programmare con Fortran versioni 90952003 Autore: Ciaburro Giuseppe

http://www.ibs.it/code/9788882336882/ciaburro-giuseppe/programmare-con-fortran.html
More information can be found here.

1.11 Japanese

Fortran 90 Explained - Metcalf and Reid, translated by H. Nisimura, H. Wada, K. Nishimura, M. Takata, Kyoritsu Shuppan Co., Ltd., 1993, ISSN 0385-6984.

Interactive Fortran 77, A Hands On Approach - I.D. Chivers and Malcom Clark. Ellis Horwood, 1984. Keigaku Publishing Company, Tokyo.

1.12 Russian

An Explanation of the Fortran 90 Programming Language (translation of Fortran 90 Explained - Metcalf and Reid), translated P. Gorbounov, Mir, Moscow, 1995, ISBN 5-03-001426-8. Available also from Petr.Gorbounov@cern.ch.

FORTRAN 77 to Fortran 90 Tutorial - Einarsson and Shokin, Russian Academy of Sciences, Novosibirsk, 1995, ISBN 5-85826-013-6.

1.13 Swedish

Fortran 90 - en introduktion - Blom, Studentlitteratur, Lund, 1994, ISBN 91-44-47881-X.

Chapter 2

Compilers

Version 1.23 September 2022. Updated the standard conformance tables, diagnostic table and added a benchmark table. Updated various vendor entries to bring up to date. Absoft cease to trade at the end of September 2022.

Version 1.22 May 2021. Intel have made major changes to way their compilers are made available. See the Intel entry for complete details.

Version 1.21 October 2020. Nvidia has integrated the PGI compiler suite into their NVIDIA HPC SDK product. Nvidia are also contributing to the LLVM Flang project. Updated the 'Additional Compiler Information' entry in the compiler section. The Polyhedron benchmarks discuss automatic parallelisation. The fortranplus entry covers the diagnostic capability of the Cray, gfortran, Intel, Nag, Oracle and Nvidia compilers. Nvidia has integrated the PGI compiler suite into their NVIDIA HPC SDK product. Nvidia are also contributing to the LLVM Flang project.

Version 1.20 June 2020. Updated the Fortran 2008 compiler standard conformance table, and added a new table on Fortran 2018 compiler standard conformance.

Version 1.19 April 2020. Added an entry for Hewlett Packard Enterprise, who have bought Cray.

Version 1.18 April 2020. Altered the status of the Oracle compiler. Informed by Robert Corbett that

> Oracle ceased Fortran development on Sept. 1, 2017. Oracle provides gfortran for its Linux systems.

The compiler and IDE are still available.

Version 1.17 December 2019. Updated the Nag entry to have details of the latest release - 7.0, which now has coarray and submodule support. See the Nag entry for more details. Intel released Parallel Studio XE 2020 in December. Added a section on compiler comparisons. Nag is the top rated diagnostic compiler.

Version 1.16 September 2019. Updated the Arm and NEC entries in the Fortran 2003 compiler conformance table. Updated gfortran entries in the Fortran 2008 compiler conformance table.

Version 1.15 August 2019. Moved several compilers to the discontinued section. Updated the Fortran 2003 compiler standard conformance data.

Version 1.14 November 2018. Updated the Fortran 2008 compiler standard conformance data.

version 1.13 October 2018. Added an Flang entry. Added details of PGI Community Edition under free compilers. Added details of Flang under free compilers. Updated the Oracle entry. Corrected compiler conformance details for some of the compilers. Moved Pathscale to no longer available.

Version 1.12, April 2018. Added an entry for SimplyFortran. Updated the compiler section to have summary details of standard conformance.

Version 1.11, April 2018. Updated the Nag entry. Added the Arm entry.

Version 1.10, March 2017. Updated to have details of the Oracle 12.6 beta version.

Version 1.9, December 2016; Updated the gfortran entry to include details of using the Windows subsystem for Linux option. Thanks to Paul Richard Thomas for the post to comp.lang.fortran bringing this to our attention

Version 1.8, June 2016; Updated the Pathscale entry

Version 1.17, May 2016; Minor edits, added the Oracle 12.5 beta.

Version 1.16, April 2016, Updated the NAG and Oracle entries.

Version 1.15, July 2015. Updated the Cray entry.

Version 1.14, June 2015. Added coverage of free compilers, compilers free for personal use, and Intel's free compiler offer for educational users.

Version 1.13, May 2013. Added an entry for Lahey GNU Shasta compiler. Thanks to Polyhedron Software for this information.

Version 1.12, January 2013. Added entry for OpenUH. Minor editorial changes. Thanks to Anton Shterenlikht for these updates and corrections.

Version 1.11, January 2013. Added entry for Nocturnal Aviation Software.

Version 1.10, August 2012. Updated Nag entry. 5.3 release supports OpenMP. Also updated product availability entry.

Version 1.9, November 2010. Updated Absoft in response to an email from Wood Lotz.

Version 1.8, July 2010. Updated Sun to reflect takeover by Oracle. Updated Salford to reflect Silverfrost rebranding.

Version 1.7, June 2009. Updated the Cray and Intel entries. Added a new section on compilers that are no longer available. This has involved moving the entries on Apogee, Compaq and NA Software into this section. For historical completeness we've also added an entry for EPC to this section.

Version 1.6, September 2008. Updated the NAG entry with more details of the Fortran Builder IDE.

Version 1.5, September 2007. Updated Absoft Entry, Compaq, Fortran Company, Fujitsu, Gnu Fortran 95, G95, IBM, Intel, Lahey/Fujitsu, NAG, NA Software, NEC, Pathscale, PGI, Salford/Silverfrost, SGI, Sun.

2.1 Introduction

The following is a list of companies and organisations that provide Fortran compilers that conform to the Fortran 90, 95, 2003 and 2008 standards. Fortran Forum has a more or less standing table on compilers that support features from the 2003 and 2008 standards and TS

2.1.1 Free compilers for general use

The following compilers are free for general use. We have included short extracts from their web sites.

Flang is a Fortran compiler targeting LLVM. Flang single-core and OpenMP performance is now on par with GNU Fortran. Flang has implemented Fortran 2003 and has a near full implementation of OpenMP through version 4.5 targeting multicore CPUs.

g95 - G95 is a stable, production Fortran 95 compiler available for multiple cpu architectures and operating systems. Innovations and optimizations continue to be worked on. Parts of the F2003 and F2008 standards have been implemented in g95.

gfortran - Gfortran is the name of the GNU Fortran project, developing a free Fortran 95/2003/2008 compiler for GCC, the GNU Compiler Collection.

HP - Only for OpenVMS. Last update 2007. Requires an Alpha or Itanium processor.

Intel - for Windows, Linux and macOS. Please see the Intel entry for full details

Nvidia NVIDIA HPC SDK. C, C++ and Fortran compilers are available.

Oracle Developer Studio 12.6; High performance C, C++ and Fortran compilers optimized for SPARC and x86 platforms. Provides perpetual no-cost license for production use and the development of commercial applications.

Visit

`https://www.oracle.com/technetwork/server-storage/developerstudio/overview/index.html`

for more information.

2.1.2 Free for personal use

The following compilers are free for personal use. License terms have been included.

Silverfrost - Silverfrost FTN95 can now be used free for personal use. It is strictly for personal use or evaluation purposes. You can use it at home on your own personal projects. You can use it to evaluate FTN95 with the intention of purchasing it later. Any applications you create with it will display a banner announcing that it is the personal edition. T his product is not directly supported but you can post questions in our forums.

2.1.3 Free for educational use

Intel - check the conditions in the Intel entry.

2.2 Standards conformance

In what follows we are referring to Fortran 90 and later standards.

2.2.1 Fortran 90 and 95 conformance

Most compilers conform to the Fortran 90 and 95 standards.

2.2.2 Fortran 2003 conformance

The Fortran 2003 standard was formally published by the ISO in November 2004. The following table

	Absoft	Arm	Cray	Fujitsu	gfortran	IBM	Intel	NAG	NEC	Oracle	PGI Nvidia
Release	14	18.1	8.5.8	2.0.3	9.3.0	15.1.5	19.1	7.0	2.3.1	8.8	Note 1
Y	37	55	55	55	57	56	52	57	57	52	55
Y Notes	0	1	1	1	0	0	4	0	0	2	1
N	15	0	0	0	0	0	0	0	0	3	0
N Notes	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	0	0	1	0	0	0	0	0
P Notes	1	1	1	1	0	0	1	0	0	0	1
Unknown	3	0	0	0	0	0	0	0	0	0	0
Total	57	57	57	57	57	57	57	57	57	57	57
	Absoft	Arm	Cray	Fujitsu	gfortran	IBM	Intel	NAG	NEC	Oracle	PGI Nvidia
Years	NA	13	4	13	15	6	9	10	14	NA	11

The *Years* figure is the time taken to fully implement the standard.

Fortran 2003 compiler standard conformance summary by vendor

has details of conformance. The information has been taken from our compiler conformance tables.

Note 1: The version number for the PGI compiler is 18.1, and for the Nvidia HPC compiler is 20.7-0.

2.2.3 Fortran 2008 conformance

The following table

Vendor	Absoft	Arm	Fujitsu	gfortran	HPE	IBM	Intel		NAG	NEC	PGI
							ifort	ifx			Nvidia
Release	14	19.1	4.0.2	9.3.0	8.5.8	15.1.5	2021.6	2022.1	7.1	2.3.1	Note 1
Y	12	28	78	48	66	53	78	77	79	79	41
Y, notes	1	0	1	1	0	0	0		0	1	1
N	35	39	35	13	0	22	0	1	0	12	24
N, notes	0	0	1	1	0	0	0		0	0	0
P	0	0	2	2	0	0	0	0	0	0	0
P, notes	0	6	2	2	0	1	0		0	0	1
No info.	31	6	12	12	13	3	1	1	0	12	12
Total	79	79	79	79	79	79	79	79	79	79	79

Fortran 2008 compiler standard conformance summary by vendor

provides details of 2008 support. The information has been taken from our compiler conformance tables.

Note 1: The version number for the PGI compiler is 18.1, and for the Nvidia HPC compiler is 20.7-0.

2.2.4 Fortran 2018 conformance

The following table

Vendor	Absoft	Arm	Fujitsu	gfortran	HPE	IBM	Intel		NAG	NEC	PGI
							ifort	ifx			Nvidia
Release	14	18.1	4.0.2	9.3.0	9.1.0	15.1.5	2021.6	2022.1	7.0	2.3.1	Note 1
Y	0	0	14	0	82	0	100	78	68	68	5
Y, notes							4	2	4		
N	0	0	88	0	22	0	0	24	32	36	0
N, notes									1		
P	0	0	1	0	0	0	0	0	0	0	0
P, notes									1		
No info.	104	104	1	104	0	104	0	0	0	0	99
Total	104	104	104	104	104	104	104	104	104	104	104

Fortran 2018 compiler standard conformance summary by vendor

provides details of 2018 support. The information has been taken from our compiler conformance tables.

Note 1: The version number for the PGI compiler is 18.1, and for the Nvidia HPC compiler is 20.7-0.

2.2.5 fortranplus compiler conformance tables

We make available detailed compiler conformance tables. Visit

`https://www.fortranplus.co.uk/fortran-information/`

for links to current and previous editions.

The ACM publication Fortran Forum also has copies.

`https://www.acm.org/special-interest-groups/sigs/fortran-forum`

Their digital library is available at

`https://dl.acm.org/`

Revision 26 is available at

`https://dl.acm.org/doi/pdf/10.1145/3345502.3345505`

2.3 Absoft

`http://www.absoft.com/`

Absoft ceased to trade in September 2022. Access to service packs and electronic delivery packages will be available until September 30th.

After September 30th – Absoft will be closed, but the User Forums are scheduled to remain active though the end of the year.

Here are some details taken from the Absoft site.

Absoft's Pro Fortran tool suite automates building extremely fast parallel code on multi-core systems. Compiler features include: APO, IPO, PGFDO optimizers and auto vectorization, Absoft's exclusive Dynamic AP load balancing technology - performance increases up to 20% on multi-core systems, SMP graphical code analyzer and OpenMP 3.0 support, Absoft Window Environment (AWE), External Libraries and Tools Plug-in, expanded F2003 & F2008 support and enhanced Fx3 graphical debugger.

Available for Windows, Linux, MacOS Intel and PPC.

2.4 Arm

Visit

`https://www.arm.com/products/development-tools/hpc-tools/allinea-studio/fortran-compiler`

for more details.

Here are some details from their site.

We work with many Arm hardware vendors to ensure that Arm Fortran compiler generates optimal code utilizing the salient features of the hardware, allowing you to get best performance out-of-the-box.

Arm Fortran Compiler provides full support for Fortran 2003 and prior standards. It also has partial support for Fortran 2008 with a plan to add more support in the future.

Our commercial compiler is based on two community-driven projects LLVM for overall compiler infrastructure and Flang for Fortran front end. This allows our users to benefit from advances from these projects in addition to specific tuning by Arm for our architecture.

Arm Fortran Compiler supports: Fortran 2003 and prior standards; Partial support for Fortran 2008; OpenMP 3.1; 64-bit Arm platforms including Cavium ThunderX2 and Qualcomm Centriq; Full support for SVE, an Arm architecture extension suited for HPC; Leading Linux distributions including Red Hat 7.3+, SLES 12+ and Ubuntu 16.04+.

2.5 Cray - now HPE

`http://www.cray.com/`

Cray has a fully optimizing Fortran 2008 compiler available for Cray XE, XK, and XC systems. The compiler supports OpenMP, OpenACC, and the enhanced features for inter-operating with C described in TS 29113.

See the HPE section also.

2.6 Flang

Flang is a Fortran compiler targeting LLVM.

`https://github.com/flang-compiler/flang`

Flang was announced in 2015. In 2017, the source code was released on GitHub.

Flang is a Fortran language front-end designed for integration with LLVM and the LLVM optimizer.

Flang+LLVM is a production-quality Fortran solution designed to be co-installed and is fully interoperable with Clang C++.

Flang single-core and OpenMP performance is now on par with GNU Fortran. Flang has implemented Fortran 2003 and has a near full implementation of OpenMP through version 4.5 targeting multicore CPUs.

The goals of Flang are:

> Ensure Flang becomes a self-sustaining open source project
>
> Attract additional developers from the (broad) community to create a critical mass of contributors
>
> Deliver single-core CPU performance comparable or better than that of gfortran
>
> Enable multicore CPU and GPU programming with a robust and performant implementation of OpenMP
>
> Create a source base that can be readily re-hosted and re-targeted to future systems as easily as Clang and LLVM
>
> Create a source base in which researchers and developers can ramp up and be productive quickly
>
> Create a source base that is componentized to a degree that enables re-use of elements of Flang in tools projects

Here is an extract of the 2015 announcement.

> The U.S. Department of Energy's National Nuclear Security Administration (NNSA) and its three national labs today announced they have reached an agreement with NVIDIA's PGI software to create an open-source Fortran compiler designed for integration with the widely used LLVM compiler infrastructure.

> LLVM is a collection of reusable compiler and tool chain technologies with a modular design that facilitates support for a wide variety of programming languages and processor architectures. The Fortran front-end module created through this project will be derived from NVIDIA's PGI Fortran compiler, which has been used in production across a variety of high performance computing systems for more than 25 years. PGI software is a leading supplier of software compilers and tools for parallel computing,

A Windows port has been initiated by the broader Flang community.

ARM Ltd. introduced a Fortran compiler for ARM/HPC based on Flang on November 13, 2017

2.7 Fortran Company

`http://www.fortran.com/`

The Fortran Company offers F, the subset language, for Unix and Windows, some in highly optimizing versions. All of the full professional versions of the F compiler are available free by downloading them from the F anonymous ftp directory.

The Fortran Tools include a Fortran 95 compiler with a graphical user interface that runs on Linux or Windows on a CD. The CD also includes several Fortran books in PDF format and many tools, such as Matran, a matrix computation library that uses the highly tuned Atlas libraries, a plotting package, and a library of routines to create GUIs for your Fortran application programs.

2.8 Fujitsu

`http://www.fujitsu.com/global/`

Fortran 95 (Solaris) A powerful, updated development system used for FORTRAN productive applications Current version: Sun Studio 9

The ISO Fortran 95 Standard is fully supported, additionally there are enhancements for Fortran77 such as pointers/structures, binary/octal/hexadecimal constants, etc. High optimization includes automatic parallelization and OpenMP support. A Fortran runtime system optimized for UltraSPARC is now also included in the package. The development environment consists of the following components:

Workbench: An integral development environment for the C/C++ and Fortran compilers, for compiler control, program execution, debugging, performance analysis, coverage etc. with a Motif-based graphical user interface (GUI).

Visual Analyzer: A development and migration tool for C/C++ and Fortran programs. The enclosed Source Analyzer allows the static program structure and the global data relations to be visualized. It contains a class browser, cross references and a calling graph viewer.

Parallel Analyser: Consists of an integrated development environment for the OpenMP programming. It contains a manager, a debugger and a profiler.

Fujitsu also has a Fortran 95 compiler for Linux and a highly optimized, native Fortran 95 compiler, Fortran/VPP and HPF, for its VPP supercomputers.

2.9 Gnu Fortran

Visit

`http://gcc.gnu.org/wiki/GFortran`

and

`http://gcc.gnu.org/fortran/`

Gfortran is the name of the GNU Fortran project, developing a free Fortran 9520032008 compiler for GCC, the GNU Compiler Collection. The gfortran development effort uses an open development environment in order to attract a larger team of developers and to ensure that gfortran can work on multiple architectures and diverse environments.

There are several options for installing and using gfortran on a Windows platform. The most recent option for Windows 10 is to use the Windows subsystem for Linux. We are currently updating our gfortran Windows notes to include details on how to do this. Thanks to Paul Richard Thomas for the post to comp.lang.fortran for bringing this to our attention.

2.10 g95

`http://www.g95.org/`

g95 is a stable, production Fortran 95 compiler available for multiple cpu architectures and operating systems. Innovations and optimizations continue to be worked on. Parts of the F2003 standard have been implemented in g95.

2.11 Hewlett Packard

The web address that describes all of the compilers and supported hardware and operating systems is

`http://h21007.www2.hp.com/dspp/tech/tech_TechSoftwareDetailPage_IDX/1,1703,6235,00.html`

HP's Fortran products are available for multiple platforms: Windows, Tru64 UNIX AlphaServer systems, Linux AlphaServer systems, and HP OpenVMS (Alpha / VAX).

Visual Fortran for Windows
Fortran for Linux Alpha
Fortran for Tru64 UNIX Alpha
Fortran for HP OpenVMS Alpha
Fortran for HP OpenVMS Integrity
Fortran for HP OpenVMS VAX
Fortran for HP-UX

The Windows product, Compaq Visual Fortran, includes the Microsoft Developer Studio IDE which can be shared with Microsoft Visual C++. Parallel execution using OpenMP-directed decomposition or HPF is included on the Tru64 UNIX platform. On Windows NT, SMP parallel execution using directed decomposition is available through Visual. Compaq Fortran for Linux Alpha Systems is available as a free download under a Technology Enthusiast license for non-commercial use. All Compaq Fortran 95 products include the Compaq Extended Math Library of optimized scientific subroutines and the allocatable array extensions.

2.12 HPE - Hewlett Packard Enterprise

HPE have bought out Cray. The following announcement was made in May 2019.

https://www.cray.com/company/news-and-media/hpe-acquisition-press-release

They market a variety of systems including

Shasta Supercomputers

XC Series Supercomputers

CS Series Cluster Supercomputers

2.13 IBM

http://www-306.ibm.com/software/awdtools/fortran/

http://www-306.ibm.com/software/awdtools/fortran/xlfortran/features/f2003.html

XL Fortran Enterprise Edition for AIX XL Fortran Enterprise Edition for AIX provides industry-leading code optimization and tuning features, a full implementation of the OpenMP API Version 2.5, Symmetric Multiprocessing (SMP) APIs, direct manipulation of the floating-point status and control register, 64-bit enablement, asynchronous I/O, debug memory routines, and many other features.

XL Fortran Advanced Edition for Blue Gene XL Fortran Advanced Edition for Blue Gene is the latest addition to our XL Fortran compiler family. It expands our proven XL Fortran compiler technology to exploit the capabilities of the PowerPC 440 and 440d processors used in IBM Blue Gene/L supercomputers.

XL Fortran Advanced Edition for Linux XL Fortran Advanced Edition for Linux supports your choice of RHEL4, SLES9, and Y-HPC Linux distributions. Advanced optimization technology and VMX support help you create high-performance 32-bit and 64-bit applications that run efficiently on a variety of processor architectures, including IBM's newest POWER5+ and PowerPC 970 processors, and Apple Power Mac G5 and Xserve G5 systems.

VS FORTRAN VS FORTRAN contains features geared to help Fortran programmers develop applications more easily and efficiently, while using the full power of IBM's large systems.

The second web link has details of Fortran 2003 support.

2.14 Intel

Their web site is:

https://software.intel.com/content/www/us/en/develop/tools/oneapi/all-toolkits.html#gs.zpo46w

Intel have changed the way they make their compilers and developer products available.

Intel now offers four different oneAPI Toolkit variants:

Intel oneAPI Base Toolkit: Get started with this foundational kit that enables developers of all types to build, test, and deploy performance-driven, data-centric applications across CPUs, GPUs, and FPGAs.

Intel oneAPI Base & HPC Toolkit: Deliver fast C++, Fortran, OpenMP, and MPI applications that scale.

Intel oneAPI Base & IoT Toolkit: Build high-performing, efficient, reliable solutions that run at the network's edge.

Intel oneAPI Base & Rendering Toolkit: Create high-performance, high-fidelity visualization applications.

The following table shows what is available in each toolkit.

Intel oneAPI Component	Base	HPC	Iot	Rendering
DPC++ / C++ Compiler	X	X	X	
DPC++ Library	X			
DPC++ Compatibility Tool	X			
GDB (Debugger)	X			
Threading Building Blocks	X			
VTune Profiler	X			
Advisor	X			
Data Analytics Library	X			
Deep Neural Networks Library	X			
Collective Communications Library	X			
Integrated Performance Primitives	X			
Integrated Performance Primitives Cryptography	X			
Math Kernel Library	X			
Video Processing Library	X			
Distribution for Python	X			
Cluster Checker		X		
Fortran Compiler		X		
Fortran Compiler Classic		X		
Inspector		X	X	
MPI Library		X		
Trace Analyzer and Collector		X		
Eclipse IDE			X	
IoT connection tools			X	
Linux Kernel Build Tool			X	
Embree				X
Open Image Denoise				X
OpenSWR				X
Open Volume Kernel Library				X
OSPRay				X

Intel prouduct availability

For general Fortran and parallel programming we recommend installing the base and hpc kits. Accepting the defaults with both kits makes two Fortran compilers available (the classic ifort compiler and the new ifx compiler, which is a beta product). You also

get the ability to do MPI programming and C and C++ programming with the default install.

The kits are available with support and pricing can be found at the Intel site, and third party suppliers.

The following is a link to Polyhedron Software and Services a UK company that offers a range of Intel products.

`https://polyhedron.com/?product_cat=intel`

A new commercial licence is 1,180 at this time. Annual maintenance (or renewal) is 420 at this time.

The toolkits are also available at no cost with support via on line user forums.

2.15 Lahey/Fujitsu

`http://www.lahey.com/`

Lahey/Fujitsu Fortran 95 is produced by the Lahey/Fujitsu alliance. LF95 is available in three Windows configurations: Express, Standard, and PRO, and two Linux configurations: Express and PRO. All configurations feature: VAX, IBM, and POSIX language extensions, allocatable array enhancements, etc. The Windows and Linux Express version is command line only and features the compiler, linker and debugger. PRO for Windows adds a Fortran-smart Windows editor, a debugger, an AUTOMAKE make utility, and an enhanced Winteracter Starter kit (WiSK) for creating true Windows programs with Fortran, and a Coverage Analysis Tool that detects unexecuted code and performs range of operation checking. The PRO is compatible with Visual C++, Visual Basic, and Delphi and also includes Fujitsu's SSL2 Math Library and Visual Analyzer (see below). The PRO Linux version offers auto-parallelization, OpenMP compatibility, thread-safe BLAS and LAPACK, WiSK, AUTOMAKE, and Fujitsu's SSL2. All products come with free technical support and are available at:

`http://www.lahey.com/`

Also available is a subset compiler, elf90.

It would appear that the Windows version is no longer under active development. The last update was dated December 2004.

2.16 Lahey/GNU Shasta Compiler

`http://www.lahey.com/`

Full Fortran 95/90/77 compliance with extensive support for the Fortran 2003 and 2008 standards; targets 32 and 64-bit Windows. Includes the automatic-parallelizing GFortran compiler, Visual Studio 2012 Shell, Lahey's Exclusive Visual Studio Fortran support, Winteracter WiSK Graphics package, and more! Compatible with Windows 8/7 (32 and 64-bit) and more!

2.17 NAG

`http://www.nag.co.uk/nagware.asp`

Nag announced the 7.0 release at SuperComputing 2019.
Major new features include

Submodules

IEEE half precision floating-point

Parallel execution of coarray programs on SMP systems

Much of the Fortran 2018 extra coarray features, in particular, teams and events.

The following is taken from the Nag site.

The NAG Fortran Compiler is robust, highly tested, and valued by developers all over the globe for its checking capabilities and detailed error reporting. The NAG Fortran Compiler is available on Linux, Microsoft Windows and Mac OS X. The latest release has extensive support for both modern and legacy Fortran features, and also supports parallel programming with OpenMP.

Cleaner code = fewer mistakes

My programs are much cleaner thanks to the new features in the NAG Fortran Compiler. This may seem like a minor issue, but there are literally hundreds of routines in my programs which pass arrays between them, and simpler interfaces lead to fewer mistakes. It's also useful to be able to write subroutines which decide the size of their return arrays internally, and to have allocatable arrays as structure components. Ian Thompson, Research Associate at Loughborough University

OpenMP 3.1 and an integrated suite of tools – it's all in there

The NAG Fortran Compiler provides support for Fortran 2008 (almost all), Fortran 2003 (complete), Fortran 95 and OpenMP 3.1. All platforms include supporting tools for software development: source file polishers, dependency generator for module and include files, call-graph generator, interface builder and a precision unifier.

It is available on Linux, Microsoft Windows and Mac OS X platforms. For users preferring an Integrated Development Environment (IDE) on Microsoft Windows or Apple Mac, NAG has developed NAG Fortran Builder.

`http://www.nag.co.uk/nagware/np/fortranbuilder.asp`

2.18 NEC

`http://www.nec.com/`

NEC has a native, optimizing Fortran 95 compiler, FORTRAN90/SX, with an automatic vectorization and parallelization capability, for its supercomputer SX series. HPF/SX V2 provides functions conforming to the specification of HPF1.1 and HPF2.0 and can be used with vector processing functions in SX Fortran and with parallel processing functions using microtasking.

2.19 Nvidia

Nvidia are involved in two Fortran compiler projects.

Here is an extract about these projects from an email from Nvidia regarding support for the Fortran 2018 standard.

> PGI/NVIDIA support for Fortran 2018 will continue to be somewhat limited in our existing compiler.
>
> As you may know, we have contributed a new front-end for Fortran 2018 to the LLVM project. This (confusingly called "Flang" like the previous open source effort) will be the basis of NVIDIA's F2018 support going forward.

2.19.1 Nvidia/PGI project

In 2020 NVIDIA integrated the PGI technology into a new NVIDIA HPC SDK product, and NVIDIA retired the "PGI Compilers and Tools" brand name.

Visit

`https://developer.nvidia.com/hpc`

for more information.

The following information has been taken from that site.

Build Scalable GPU-Accelerated Applications. Faster. Researchers, scientists, and developers are advancing science by accelerating their high-performance computing (HPC) applications on NVIDIA GPUs, which have the computational capacity to tackle today's most challenging scientific problems. From computational science to AI, GPU-accelerated applications are delivering groundbreaking scientific discoveries. And popular languages like C, C++, Fortran, and Python are being used to develop, optimize, and deploy these applications.

NVIDIA GPUs can be programmed much like CPUs. Start by substituting GPU-optimized math libraries. Add additional acceleration using the standard C++ parallel algorithms and Fortran language features. Use pragmas and directives to fill any standard language gaps, and finally, optimize performance with CUDA.

The NVIDIA HPC SDK

A Comprehensive Suite of Fortran, C, and C++ Development Tools and Libraries The NVIDIA HPC SDK is a comprehensive toolbox for GPU accelerating HPC modeling and simulation applications. It includes the C, C++, and Fortran compilers, libraries, and analysis tools necessary for developing HPC applications on the NVIDIA platform. Use the NVIDIA HPC SDK to maximize your productivity and the performance and portability of your code.

The NVIDIA HPC SDK C, C++, and Fortran compilers support GPU acceleration of HPC modeling and simulation applications with standard C++ and Fortran, OpenACC directives, and CUDA . GPU-accelerated math libraries maximize performance on common HPC algorithms, and optimized communications libraries enable standards-based multi-GPU and scalable systems programming. Performance profiling and debugging tools simplify porting and optimization of HPC applications, and containerization tools enable easy deployment on-premises or in the cloud. With support for NVIDIA GPUs and Arm, OpenPOWER, or x86-64 CPUs running Linux, the HPC SDK provides the tools you need to build NVIDIA GPU-accelerated HPC applications.

Here is the download page.
`https://developer.nvidia.com/nvidia-hpc-sdk-download`
Versions are currently available for

Linux X86_64

Linux OpenPOWER

Linux Arm Server

Windows x64 - under development

Below is a list of the PGI compilers that have been rebranded and integrated into the NVIDIA HPC SDK.

Fortran: nvfortran - formerly pgfortran

C: nvc - formerly pgcc

C++: nvc++ - formerly pgc++

2.19.2 LLVM and Flang project

Here is the base address for this work.
`https://github.com/flang-compiler/flang`
Here is an extract from that site.

Flang is a Fortran compiler targeting LLVM.

Visit the flang wiki for more information:
`https://github.com/flang-compiler/flang/wiki`

2.20 OpenUH

`http://www2.cs.uh.edu/~openuh/`
The following is taken from their site.

OpenUH is an open source, optimizing compiler suite for C, C++ and Fortran 95. It supports a variety of architectures including IA-32, X86_64, IA-64. To achieve portability, OpenUH is able to emit optimized C or Fortran 77 code that may be compiled by a native compiler on other platforms. The supporting runtime libraries are also portable - the OpenMP runtime library is based on the portable Pthreads interface while the Coarray Fortran runtime library is based, optionally, on the portable GASNet or ARMCI communications interfaces. OpenUH includes support for a new version of the Dragon tool that gathers and displays static and dynamic information about a user's application.

2.21 Oracle - originally Sun

Available, but no longer developed.

Informed by Robert Corbett that

> Oracle ceased Fortran development on Sept. 1, 2017. Oracle provides gfortran for its Linux systems.

`https://www.oracle.com/technetwork/server-storage/developerstudio/overview/index.html`

2.21.1 Mixed-Language Development

C, C++ Compilers

Optimize application performance on the latest Oracle systems, on-premise and in the cloud

Fortran Compiler

Optimizes compute-intensive application performance

Debugger

Ensures application stability with event handling and multi-thread support

Performance Library

Maximizes compute-intensive application performance using advanced numerical solver libraries

2.21.2 Application Analytics

Performance Analyzer

Provides deep insight into C, C++, Java, Scala, and Fortran application performance, allowing you to easily identify bottlenecks and tune for optimal performance

Code Analyzer

Protects your application from security vulnerabilities, and includes SPARC Software in Silicon support to secure your application at record speeds

Thread Analyzer

Detects hard to pinpoint race and deadlock conditions in multi-threaded applications

2.21.3 IDE

Oracle Developer Studio IDE

Extensible, full-featured, and simplifies cloud development for C, C++ and mixed C++/Java applications

Download from:

`https://www.oracle.com/technetwork/server-storage/developerstudio/downloads/index.html`

2.22 PGI

PGI (formerly The Portland Group, Inc.) was a company that produced a set of commercially available Fortran, C and C++ compilers for high-performance computing systems.

On July 29, 2013, NVIDIA Corporation acquired The Portland Group, Inc.

As of August 5, 2020, the PGI Compilers and Tools technology is a part of the NVIDIA HPC SDK product available as a free download from NVIDIA.

See the Nvidia entry for details of the current offerings.

2.23 Silverfrost, nee Salford Software

`http://www.silverfrost.com/11/ftn95/overview.aspx`

Salford Software markets FTN95, a Fortran 95 compiler for Win32, running on Windows 95/NT/2000/XP PCs. It has announced its Fortran 95 compiler for Microsoft .NET (FTN95 for .NET). This compiler will produce fast executables from source files that may be any combination of Fortran 77, Fortran 90 and Fortran 95. FTN95 for .NET, including integrated Help and Debugger, is supplied bundled with FTN95 for Win32 and, optionally, with Microsoft Visual Studio for .NET. A low-cost, fully-featured personal edition is also available.

2.24 SimplyFortran

The following has been taken from the SimplyFortran site.

`http://simplyfortran.com/`

Simply Fortran - A modern Fortran development environment for Microsoft Windows, Apple macOS, and GNU/Linux systems.

Windows package includes a complete Fortran compiler, a graphical interface library, and an integrated debugger for fast and easy installation

Simply Fortran has been designed from the start with GNU Fortran integration as the primary goal. Installing the development environment is as simple as downloading and running the installer; no additional download or user configuration is necessary.

2.25 Sun - see Oracle

2.26 No longer available

2.26.1 Apogee

`http://www.apogee.com/`

Features of the FORTRAN 77/90 Compiler

No longer available. Originally available for the Solaris/SPARC platforms, the compiler conforms to Sun's Solaris ABI and produces assembler code files acceptable to Sun's Solaris assembler. When used in the FORTRAN 77 compilation mode, the compiler is

compliant with the MIL-STD 1753 FORTRAN 77 and accepts most FORTRAN 77 extensions of Sun, IBM, and other F77 compilers. The supported F77 extensions include structures, length qualification on types, additional data and constant types, initializations in type statements, additional statements (END DO, DO WHILE, POINTER, VOLATILE, etc.), computations with aggregates, namelist-directed I/Os, and debugging statements.

When used in the Fortran 90 mode, the compiler is compliant with the ANSI/ISO Fortran 90 standard.

2.26.2 Compaq

This compiler is no longer under development. This ceased when Intel bought out the technology from HP. Still widely used with legacy software. Copies for sale can be found on Ebay and similar sites.

2.26.3 EPC

Edinburgh Portable Compilers was an early vendor to produce a Fortran 90 compilation system. A report by Adam Marshall from Liverpool University has a comparison of several early Fortran 90 compilers. It can be found at the address below.

`http://www.liv.ac.uk/HPC/FortranCompilerStudyHTML/FortranCompilerStudyHTML.html`

2.26.4 NA Software

`http://www.nasoftware.co.uk/home.html`

No longer available.

2.26.5 Nocturnal Aviation Software

`http://www.nocturnalaviationsoftware.com/`

`http://www.nocturnalaviationsoftware.com/FTranProjectBuilder/`

The following is taken from their site.

Nocturnal Aviation Software is an indie Mac software developer located in Tallahassee, FL. We have almost 40 years of Fortran programming experience, 26 years of Mac user experience and have been coding for the Mac since the OS X public beta, a decade ago. We wrote these Apps to use in our own programming work because nothing else like them is available. We'd like to say "Providing software solutions since 2011", but that won't sound too impressive for quite a few years. One has to start somewhere.

No longer appears to be available.

2.26.6 PathScale

`https://en.wikipedia.org/wiki/PathScale`

The following was taken from the Pathscale site.

PathScale's goal is to make it easier to develop and deploy 64-bit applications into clustered environments. PathScale has developed one of the industry's highest-performance C, C++, and Fortran compilers for 64-bit Linux based systems.

They offer 4 products. The content and descriptions are taken from the Pathscale site.

The EKOPath 6 compiler suite (C, C++, Fortran)

ENZO 2015 - a complete GPGPU and multi-core solution, which tightly couples the best programming models with highly optimizing code generation for NVIDIA Tesla and AMD FirePro discrete GPU.

ARMv8 - ARMv8 is an emerging architecture that promises to deliver better performance per watt than your existing HPC and Enterprise hardware solutions for your applications

DF IDE - DogFood (DF), the intelligent C++/Fortran IDE that delivers productivity without getting in the way of writing, reading and navigating code. DF is not afraid to break the rules and attempt to raise the bar on what an IDE should be. The engineers who work on DF are the engineers who also use DF.

The compiler is available as a free nightly download.

2.26.7 SGI

`http://www.sgi.com/products/software/irix/tools/fortran.html`

The following is taken from the SGI site.

MIPSpro Fortran Compilers

This 64-bit ANSI Fortran 77 compiler is ideal for systems running IRIX 6.x. It is compatible with VAX/VMS Fortran and supports Cray extensions. The 7.4 version of the MIPSpro Fortran 77 and Fortran 90 compilers now support the OpenMP 2.0 standard. Among the new features introduced in the OpenMP 2.0 specification are:

WORKSHARE directive

COPYPRIVATE clause for the broadcast of sequential reads

Portable timing routines

MIPSpro Fortran 90 Compiler

A 64 bit ANSI Fortran 90 compiler with additional support for user-defined multiprocessing directives for systems running IRIX 6.x. Compatible with VAX/VMS Fortran and supports Cray extensions. With the release of version 7.4, Fortran 90 specific support under the OpenMP 2.0 standard are:

Parallelization of F90 array syntax via the WORKSHARE directive

Privatization of deferred shape and assumed shape objects

The full Fortran 2.0 specification can obtained from the OpenMP Web site at:
`http://www.openmp.org`
For more information, read about the MIPSpro compilers.
No longer available.

2.27 Additional compiler information

There are a number of sites that have additional compiler information.

2.27.1 Compiler benchmarking - Polyhedron Software and Services

Polyedron Software and Services have benchmarks tables for Linux and Windows based compilers.

The home address is
`https://polyhedron.com/`
Here is an extract from their site.

> Throughout, the optimizers of the compiler systems were used to get the "highest optimization level" (according to the compiler's documentation) and – if supported – processor-specific options were set, so that the binary code for the existing CPU was optimized. Furthermore, if not specified by the optimization level, loop unrolling was used.
>
> AP stands for AutoParallel, i.e. the compiler system also tries to parallelize the code automatically.
>
> The benchmark programs were run several times, at least 10 times in a row, and the respective run-time was measured. If the run-time measurements stabilized (i.e., was less than 1% deviation from the previous measurement), the benchmark for the affected program was completed.

Visit their site for complete details.

2.27.2 Compiler comparisons - Polyhedron Solutions

Their home address is
`https://www.fortran.uk/`
and the tables can be found at
`https://www.fortran.uk/fortran-compiler-comparisons/`
There are three tables per operating system.

Windows Compilers

- Language – What extensions to standard Fortran do the compilers support?
- Diagnostic Capabilities – Can they spot programming errors?
- Fortran Execution Time Benchmarks (64-bit Windows 7 on Intel processor) – 17 Fortran benchmarks

Linux Compilers

- Language – What extensions to standard Fortran do the compilers support?

– Diagnostic Capabilities – Can they spot programming errors?

– Fortran Execution Time Benchmarks (Linux64 on Intel processor) – 17 Fortran benchmarks

Historic benchmarks for 2015 and 2014 are also available.
`https://www.fortran.uk/historic-tables/`

2.27.3 Compiler diagnostic capability - fortranplus

We have used the Polyhedron Bench suite to produce up to date diagnostic ratings for the Cray, gfortran, Intel, Nag, and combined PGI Nvidia compilers.

Here is a summary table with diagnostic scores.

Table 2.1: Diagnostic compiler test results

Vendor	Cray	gfortran	Intel		NAG	PGI
			ifort	ifx		Nvidia
Version	10.0.4	12.2.0	2021.1	2022.1.0	7.1	22.5
Score	49	42	55	28	96	23

Visit the fortranplus site for the complete tables.

Nag has the highest diagnostic rating.

2.27.4 Compiler benchmarking - fortranplus

We have used the Polyhedron Bench suite to produce up to date benchmark ratings for the compilers we currently use on a PC platform:

NAG

Intel

gfortran

Nvidia

on the following operating systems

Windows 10

openSuSe Leap 15.3

openSuSe Tumbleweed

Redhat 9

Ubuntu 20.04.4

using the following hardware

Dell Precision 5820 workstation, Intel I9 10980 XE processor, 32 GB RAM, 18 cores, 36 with hyper threading.

Dell Vostro 5515 laptop, AMD Ryzen 7 5500U processor, 32 GB RAM. 8 cores, 16 with hyper threading.

Here are the hardware and operating system combinations.

Dell 5515, Windows 10, native

Dell 5280, Windows 10, native

Dell 5515, openSuSe Leap 15.3, hyper-v

Dell 5280, openSuSe Leap 15.3, hyper-v

Dell 5515, openSuSe Tumbleweed, WSL

Dell 5280, openSuSe Tumbleweed, WSL

Dell 5280, Redhat 9, hyper-v

Dell 5280, Ubuntu 20.04.4, Native

Notes	gfortran	ifort	ifort parallel	ifx	nag	nvidia	nvidia parallel	OS Type
Dell 5515								
1	16.41	18.41	16.44	16.60	18.51	NA	NA	Native
2	NR	13.83	15.59	7.09	19.09	9.17	13.48	hyper-v
3	14.43	14.80	17.52	7.81	NI	12.18	12.61	WSL
Dell 5820								
4	2.76	11.16	11.61	11.30	17.48	NA	NA	Native
5	13.19	9.02	8.11	7.05	NI	9.97	14.55	Native
6	13.08	8.50	7.39	6.24	19.09	9.17	12.61	hyper-v
7	13.07	10.45	9.14	10.44	NI	9.71	14.83	WSL
8	12.91	8.53	7.53	6.15	NI	9.58	12.65	hyper-v

Polyhedron compiler benchmarks

```
Notes
1       AMD, Windows 10, native, gfortran 12.1.0
2       AMD, hyper-v, openSuSe 15.3, gfortran 7.5.0
3       AMD, WSL, openSuSe Tumbleweed, gfortran 12.1.0
4       Intel, Windows 10, native, gfortran 12.1.0 (equation.com)
5       Intel, Ubuntu 20.04.4, native, gfortran 9.1.0
6       Intel, hyper-v, openSuSe 15.3, gfortran 7.5.0
7       Intel, WSL, openSuSe Tumbleweed, gfortran 12.1.0
8       Intel, hyper-v, Redhat 9, gfortran 11.2.1
Native  Native Windows (Windows 10) or
        linux (Ubuntu 20.04.4) operating system
NA      No Nvidia HPC Toolkit for Windows
NI      Not installed
NR      gfortran is not in the openSuSe repository
```

Visit the fortranplus site for the complete tables.

Chapter 3

Numeric libraries and software repositories

Version 1.01, renamed to numeric libraries and software repositories. Added details of software made available by NASA

A numeric library is set of functions and subroutines used in software development for solving numeric problems.

It is normally in one of two forms

Source form

or

Compiled form

Libraries available in source form (Fortran 90 or Fortran 77) are free.

Compiled libraries are generally commercial.

Collaboration is also quite common in the academic and science and engineering areas.

The BLAS and LAPACK libraries have been written by people working in the linear algebra area.

Cern and Harwell and have developed libraries for use in their institutions and have made their libraries (CERNLIB and HSL) publically available.

Nag stands for Numerical Algorithms Group and their major area of expertise is in the numeric area. They provide several commercial libraries. They also provide numeric consultancy services.

IMSL is another commercial library and this is sold by Roguewave Software and Absoft.

The Intel Math Kernel Library is a library of optimized math routines for science, engineering, and financial applications. Core math functions include BLAS, LAPACK, ScaLAPACK, sparse solvers, fast Fourier transforms, and vector math. It is bundled with their Fortran and C++ compilers and can also be bought separately for the Windows platform. It is currently free under Linux. It targets Intel processors.

Other publically available libraries include

ARPACK

Lis

MINPACK

Novas

PAW

PETSc

QUADPACK

SLATEC

SOFA

The Netlib site hosts a lot of useful software.
NASA also host a software repository.
More detailed information can be found in the sections that follow.

3.1 ARPACK

ARPACK is a collection of Fortran77 subroutines designed to solve large scale eigenvalue problems. The package is designed to compute a few eigenvalues and corresponding eigenvectors of a general n by n matrix A. It is most appropriate for large sparse or structured matrices A where structured means that a matrix-vector product w ¡- Av requires order n rather than the usual order n2 floating point operations. This software is based upon an algorithmic variant of the Arnoldi process called the Implicitly Restarted Arnoldi Method (IRAM). When the matrix A is symmetric it reduces to a variant of the Lanczos process called the Implicitly Restarted Lanczos Method (IRLM). These variants may be viewed as a synthesis of the Arnoldi/Lanczos process with the Implicitly Shifted QR technique that is suitable for large scale problems. For many standard problems, a matrix factorization is not required. Only the action of the matrix on a vector is needed.

ARPACK software is capable of solving large scale symmetric, nonsymmetric, and generalized eigenproblems from significant application areas. The software is designed to compute a few (k) eigenvalues with user specified features such as those of largest real part or largest magnitude.

Storage requirements are on the order of n*k locations. No auxiliary storage is required. A set of Schur basis vectors for the desired k-dimensional eigen-space is computed which is numerically orthogonal to working precision.

`https://www.caam.rice.edu/software/ARPACK/`

3.2 BLAS

The BLAS (Basic Linear Algebra Subprograms) are low level routines that provide standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform

matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. Because the BLAS are efficient, portable, and widely available, they are commonly used in the development of high quality linear algebra software, e.g. LAPACK, the Nag library, The Intel MKL library.

`http://www.netlib.org/blas/`

3.3 CERNLIB

CERNLIB is a collection of FORTRAN 77 libraries and modules.

`https://cernlib.web.cern.ch/cernlib/`

3.4 Harwell

HSL (formerly the Harwell Subroutine Library) is a collection of state-of-the-art packages for large-scale scientific computation written and developed by the Computational Mathematics Group at the STFC Rutherford Appleton Laboratory and other experts. HSL offers users a high standard of reliability and has an international reputation as a source of robust and efficient numerical software. Among its best known packages are those for the solution of sparse linear systems of equations and sparse eigenvalue problems. MATLAB interfaces are offered for selected packages. The Library was started in 1963 and was originally used at the Harwell Laboratory on IBM mainframes running under OS and MVS. Over the years, the Library has evolved and has been extensively used on a wide range of computers, from supercomputers to modern PCs. Recent additions include optimised support for multicore processors.

`http://www.hsl.rl.ac.uk/`

3.5 IMSL

The IMSL Numerical Libraries are cross-platform libraries containing a comprehensive set of mathematical and statistical functions that can be embedded in a users application.

The following document

`https://www.roguewave.com/sites/rw/files/attachments/RW-IMSL-Numerical-Libraries-datasheet.pdf`

is a 2 page summary of what is available in the Fortran library. As of September 2019 the library has been compiled with Intel 17, Sun Studio 12.5 and Absoft Pro Fortran 19. The Intel version runs on Windows and Linux. The Absoft version runs on Windows, Linux and OS X 10.

`https://www.roguewave.com/products-services/imsl-numerical-libraries`

3.6 Intel MKL

Intel Math Kernel Library (Intel MKL) is a library of optimized math routines for science, engineering, and financial applications. Core math functions include BLAS, LAPACK, ScaLAPACK, sparse solvers, fast Fourier transforms, and vector math. The routines in MKL are optimized specifically for Intel processors.

The library supports Intel processors and is available for Windows, Linux and macOS operating systems.

The main Intel site is

`https://software.intel.com/mkl`

Here is a quote from the Intel site.

> Intel Math Kernel Library (Intel MKL) optimizes code with minimal effort for future generations of Intel processors. It is compatible with your choice of compilers, languages, operating systems, and linking and threading models.
>
> > Features highly optimized, threaded, and vectorized math functions that maximize performance on each processor family
> >
> > Uses industry-standard C and Fortran APIs for compatibility with popular BLAS, LAPACK, and FFTW functions—no code changes required
> >
> > Dispatches optimized code for each processor automatically without the need to branch code
> >
> > Provides Priority Support that connects you directly to Intel engineers for confidential answers to technical questions

3.7 LAPACK

LAPACK is written in Fortran 90 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision.

The original goal of the LAPACK project was to make the widely used EISPACK and LINPACK libraries run efficiently on shared-memory vector and parallel processors. On these machines, LINPACK and EISPACK are inefficient because their memory access patterns disregard the multi-layered memory hierarchies of the machines, thereby spending too much time moving data instead of doing useful floating-point operations. LAPACK addresses this problem by reorganizing the algorithms to use block matrix operations, such as matrix multiplication, in the innermost loops. These block operations can be optimized for each architecture to account for the memory hierarchy, and so provide a transportable way to achieve high efficiency on diverse modern machines. We use the term "transportable" instead of "portable" because, for fastest possible performance, LAPACK requires that highly optimized block matrix operations be already implemented on each machine. LAPACK routines are written so that as much as possible of the computation is performed by calls to the Basic Linear Algebra Subprograms (BLAS). LAPACK is designed at the outset to exploit the level 3 BLAS - a set of specifications for Fortran subprograms that do various types of matrix multiplication and the solution of triangular

systems with multiple right-hand sides. Because of the coarse granularity of the Level 3 BLAS operations, their use promotes high efficiency on many high-performance computers, particularly if specially coded implementations are provided by the manufacturer.

Highly efficient machine-specific implementations of the BLAS are available for many modern high-performance computers. For details of known vendor- or ISV-provided BLAS, consult the BLAS FAQ. Alternatively, the user can download ATLAS to automatically generate an optimized BLAS library for the architecture. A Fortran 77 reference implementation of the BLAS is available from netlib; however, its use is discouraged as it will not perform as well as a specifically tuned implementation.

`http://performance.netlib.org/lapack/`

3.8 Lis

Lis (Library of Iterative Solvers for linear systems, pronounced [lis]) is a parallel software library for solving linear equations and eigenvalue problems that arise in the numerical solution of partial differential equations using iterative methods. The installation of Lis requires a C compiler. The Fortran interface requires a Fortran compiler, and the algebraic multigrid preconditioner requires a Fortran 90 compiler. For parallel computing environments, an OpenMP or MPI library is used. Both the Harwell-Boeing and Matrix Market formats are supported to import and export user data.

`https://www.ssisc.org/lis/index.en.html`

3.9 MINPACK

Minpack includes software for solving nonlinear equations and nonlinear least squares problems. Five algorithmic paths each include a core subroutine and an easy-to-use driver. The algorithms proceed either from an analytic specification of the Jacobian matrix or directly from the problem functions. The paths include facilities for systems of equations with a banded Jacobian matrix, for least squares problems with a large amount of data, and for checking the consistency of the Jacobian matrix with the functions.

`http://www.netlib.org/minpack/`

3.10 NAG

NAG are a not for profit organisation that provide expertise in the numeric area. They provide a variety of libraries.

`https://www.nag.com/`

Here is an extract from their site.

> NAG Library algorithms - performance driven - accurate to the core. Algorithms developed to solve complex mathematical problems quickly and easily. Algorithms that are stringently tested, expertly documented, supported and continually updated with new cutting edge algorithmic functionality. Whether in business critical applications or ground-breaking research, good numerical algorithms are the difference between success or failure.

The serial version of the Nag library is available for 32 and 64 bit versions of Windows, Linux and OS X, on x86 hardware and also on Solaris (Sparc). The parallel version (SMP and multicore) is available for 32 and 64 bit versions of Linux and Windows. A version of the library is also available for the Xeon Phi under Linux.

An MPI version of the library is also available.

The Nag libraries are also callable from a variety of languages and environments including Python, Java and the Microsoft Windows .Net framework.

We've over 50 years experience of working with the Nag library (in computing services within the University of London) and found it a very reliable and useful product and it was extensively used in a wide range of departments in science and engineering at Imperial College, Chelsea College and King's College.

3.11 NASA

Visit

`https://software.nasa.gov/`

Here is some information from this site.

> The NASA software listed in the catalog is available for use at no charge. Certain codes have been licensed by NASA for commercial purposes and are only available to other agencies or companies with a government contract. If a software product is available for commercial purposes, a statement indicating that the code "is available for licensing" will be included in the software description.

Here is a selection of some of the Fortran codes available.

> Fortran Argument Parser
>
> The Parallel Fortran Logger (pFlogger)
>
> Fortran Unit Testing Framework (fUnit v1.0)
>
> Fortran Template Library (gFTL) for managing collections of objects via software containers.
>
> Basic Comparison of Python, Julia, Matlab, IDL, R, Java, Scala, Fortran and C
>
> NASA.rb (formerly fUnit) is a collection of Fortran modules that provide a framework for automating the construction, execution, and reporting of unit tests for Fortran software applications. Support is provided for several aspects of unit testing that are peculiar to scientific technical...
>
> BLAYER is a Fortran program used for calculating compressible laminar and turbulent boundary layers in arbitrary pressure gradients.
>
> The TSONIC Fortran program calculates the transonic velocity on the blade-to-blade stream surface of a turbo-machine.

Venus-GRAM 2005 is a Fortran-based program that provides engineering estimates of density, temperature, pressure, and winds for the Venus atmosphere.

CCGEOM is a Fortran computer code developed to facilitate the rapid generation of the flow passage and blading for various turbomachinery components.

MERIDLN is a Fortran program for calculating velocities/streamlines of axial-, radial-, or mixed-flow turbo-machinery or annular ducts on the hub-shroud mid-channel stream surface.

The Thermal Insulation System Analysis Tool has been updated with more test data from the Cryogenics Test Laboratory and has been converted to Fortran 95 to allow for easier distribution.

Titan-GRAM is a Fortran-based program that provides engineering estimates of density, temperature, pressure, and winds for the Titan atmosphere. More information on the Space Environments & Effects (SEE) Program can be found at `http://see.msfc.nasa.gov/`

CAPO analyzes a Fortran program and inserts OpenMP directives into the code to improve its performance on a parallel machine. The tool relies on accurate interprocedural data-dependence information currently provided by CAPTools, which was developed at the University of Greenwich.

JeNo is a Fortran 90 computer code that calculates the far-field sound spectral density produced by axisymmetric jets at user-specified observer locations and frequency ranges. The user must provide a structured computational grid and also input a mean flow solution from a Reynolds-Averaged...

GFR is a high-order computational fluid dynamics (CFD) Fortran code for large-eddy simulations. It is based on the simple and efficient flux reconstruction method and accurate to an arbitrary order through an user-supplied input parameter. It is currently capable of using unstructured grids...

Earth Gram 2010 is an open-source Fortran computer code that can run on a variety of platforms including PCs and UNIX stations. The model provides values for atmospheric parameters such as density, temperature, winds, and constituents for any month and at any altitude and location within the...

JeNo is a Fortran 90 computer code that evaluates the far-field turbulence-generated noise in non-axisymmetric jets. The propagation Green's function is calculated along a user-specified azimuthal angle (line of sight), but source volume integration is carried out in 3D. The user must provide a...

RSI is a Fortran computer code for calculating the spectrum of broadband noise produced by the interaction of fan-rotor wake turbulence with fan-exit guide vanes (i.e., the stator). Provided with incident-turbulence characteristics, the code computes the spectra of acoustic power upstream and...

3.12 Netlib

Netlib is a repository of scientific computing software which contains a large number of separate programs and libraries including BLAS, EISPACK, LAPACK and others.

`http://www.netlib.org/`

Here is a link to their FAQ.

`http://www.netlib.org/misc/faq.html#2.1`

3.13 NOVAS

The Naval Observatory Vector Astrometry Software (NOVAS) is a software library for astrometry-related numerical computations. It is developed by the Astronomical Applications Department, United States Naval Observatory. Currently, NOVAS has three different editions for C, Fortran, and Python, respectively.

`https://en.wikipedia.org/wiki/Naval_Observatory_Vector_Astrometry_Subroutines`

3.14 PAW

PAW is conceived as an instrument to assist physicists in the analysis and presentation of their data. It provides interactive graphical presentation and statistical or mathematical analysis, working on objects familiar to physicists like histograms, event files (Ntuples), vectors, etc. PAW is based on several components of the CERN Program Library. Like CERN Program Library PAW usage and/or redistribution is granted under the terms of the GNU General Public License.

`http://paw.web.cern.ch/paw/`

3.15 PETSc

PETSc, pronounced PET-see (the S is silent), is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It supports MPI, and GPUs through CUDA or OpenCL, as well as hybrid MPI-GPU parallelism. PETSc (sometimes called PETSc/Tao) also contains the Tao optimization software library.

`https://www.mcs.anl.gov/petsc/`

3.16 QUADPACK

QUADPACK is a FORTRAN subroutine package for the numerical computation of definite one-dimensional integrals. It originated from a joint project of R. Piessens and E. de Doncker (Appl. Math. and Progr. Div.- K.U.Leuven, Belgium), C. Ueberhuber (Inst. Fuer Math.- Techn.U.Wien, Austria), and D. Kahaner (Nation. Bur. of Standards-Washington D.C., U.S.A.).

`http://www.netlib.org/quadpack/`

3.17 SLATEC

SLATEC Common Mathematical Library, Version 4.1, July 1993 a comprehensive software library containing over 1400 general purpose mathematical and statistical routines written in Fortran 77.

`http://www.netlib.org/slatec/`

3.18 SOFA

The International Astronomical Union's SOFA service has the task of establishing and maintaining an accessible and authoritative set of algorithms and procedures that implement standard models used in fundamental astronomy.

The SOFA Software Collection is currently available in both Fortran 77 and ANSI C.

`http://www.iausofa.org/`

Chapter 4

Debuggers

Version 1 January 2013. Thanks to Anton Shterenlikht for the initial request for a section on debuggers. Thanks to Wood Lotz, Van Snyder, Bill Long and Tobias Burnus for their contributions.

4.1 Introduction

A number of contributors have mentioned the simple inclusion of print statements in your code - which is of course simple and very effective.

4.2 Absoft

The following information was provided by Wood Lotz.

Absoft Fx3 debugger

It supports Fortran, C and assm and has been evolving over the past 15 years. During that period it has supported and or sold with, a wide variety of compilers though currently we offer it only as a component bundled with the Absoft Pro Fortran products for Windows, Linux and Mac.

Product overview

`http://www.absoft.com/Absoft_FxDebugger.htm`

4.3 DDT

`http://www.allinea.com/products/ddt/`

The following information is taken from the above site.

Allinea DDT is the most advanced debugging tool available for scalar, multi-threaded and large-scale parallel applications. It debugs code on:

workstations GPUs clusters, and... the very largest supercomputers. Comprehensive and easy-to-use Allinea DDT has many features that are missing from ordinary debuggers — such as memory debugging and data visualization. With an acclaimed user interface that wins praise for ease-of-use and capability, it is quite simply an integral part of efficient software development.

Simplifying debugging at scale. For multi-threaded or OpenMP development, Allinea DDT enables threads to be controlled individually and collectively with advanced capabilities for comparing data across threads.

The Parallel Stack Viewer is a unique way to see the program state of all processes and threads at a glance. You can easily spot rogue processes or threads and even using it to define new control groups, making massive parallel programs far easier to manage. The Allinea DDT interface scales amazingly to provide the same clarity of information at thousands of processes as at a handful. It highlights commonality and differences using summary views and data comparisons to focus your attention.

Allinea DDT has been proven at scale on the most powerful systems - including debugging applications at over 200,000 cores simultaneously. Allinea DDT puts you in control of your application, whether you are working with a workstation or a thousand processor, high-performance cluster.

Debug code on the CPU and GPU from a single tool This powerful combination gives you all the advanced debugging features of Allinea DDT but with the enhanced ability to debug CUDA code. With a single tool, you can debug hybrid MPI, OpenMP and CUDA applications on a single workstation or GPU cluster. Features such as the detection of invalid memory accesses, the visualization of GPU data, and GPU thread control have been designed to help you find the GPU porting bugs quickly and easily.

Allinea DDT supports the NVIDIA CUDA Toolkit and is fully compatible with NVIDIA's Fermi architecture.

4.4 gnu gdb

A "-g" without further optimization option (such as -O2) usually disables optimization, which some compiler have otherwise by default. "-g" can also be used with optimization; however, the optimization makes debugging more difficult (even though compilers might try hard to keep some debugging information available).

Additionally, "-g" may or may not have an effect on code generation - especially when used together with optimization (such as -O2); some compiles generate the same code with debug information than without, others allow "-g -O2" but generate slightly different code than with only "-O2". Using "-O1 -g" is often a good compromise between performance and debuggability as it does some optimization but usually no inlining and other debugging unfriendly optimizations.

For GCC:

* GCC by default has no optimization (-O0), thus "-g" doesn't change the optimization level. * GCC generates the same code with and without "-g" for all optimization levels [if not, it is a bug] * GCC 4.8 has the new option -Og, which enables optimizations that do not interfere with debugging.

In addition, newer debug formats (DWARF, latest is DWARF4) allow for better debugging support, but require also newer debugging tools. For instance, -gdwarf-4 is the default in GCC with/since 4.8 but requires GDB 7.5, Valgrind 3.8 and elftools 0.154. GCC also supports (since 4.7) a GNU extension (supported by GDB 7.4, proposed for DWARF5) "entry value"/ "call site" which allows - with restrictions - to debug function calls where the argument has been passed in registers. (For completeness: DWARF4 is

supported - optionally - since GCC 4.6.)

4.5 Intel IDB

4.6 LLDB

4.7 Microsoft Visual Studio Debugger

4.8 totalview

Visit

`http://www.roguewave.com/products/totalview.aspx`

for detailed information.

Here is some information taken from the above site.

TotalView is a GUI-based source code defect analysis tool that gives you unprecedented control over processes and thread execution and visibility into program state and variables.

It allows you to debug one or many processes and/or threads in a single window with complete control over program execution. This allows you to set breakpoints, stepping line by line through the code on a single thread, or with coordinated groups of processes or threads, and run or halt arbitrary sets of processes or threads. You can reproduce and troubleshoot difficult problems that can occur in concurrent programs that take advantage of threads, OpenMP, MPI, GPUs or coprocessors.

TotalView provides analytical displays of the state of your running program for efficient debugging of memory errors and leaks and diagnosis of subtle problems like deadlocks and race conditions. Whether you are a scientific and technical computing veteran, or a software professional new to the development challenges of multi-core or parallel applications, TotalView gives you the insight needed to find and correct errors quickly, validate prototypes, verify calculations and certify code. TotalView works with C, C++ and Fortran applications written for Linux (including the Blue Gene platforms), UNIX and Mac OS X platforms. It includes sophisticated memory debugging and analysis, reverse debugging, Xeon Phi coprocessor and OpenACC / CUDA debugging capabilities.

A tutorial is available at

`https://computing.llnl.gov/tutorials/totalview/`

Here is some additional information from that site.

TotalView is a sophisticated and powerful tool used for debugging and analyzing both serial and parallel programs. TotalView provides source level debugging for serial, parallel, multi-process and multi-threaded codes, and can be used in a variety of UNIX environments, including those with distributed, clustered, stand-alone and SMP machines. TotalView provides both a graphical user interface and command line interface. TotalView has been selected as the Department of Energy's ASC debugger of choice for it HPC platforms.

This tutorial has three parts, each of which includes a lab exercise. Part 1 begins with an overview of TotalView and then provides detailed instructions on how to set up and

use its basic functions. Part 2 continues by introducing a number of new functions and also providing a more in-depth look at some of the basic functions. Part 3 covers parallel debugging, including threads, MPI, OpenMP and hybrid programs. Part 3 concludes with a discussion on debugging in batch mode.

Level/Prerequisites: This tutorial is one of the eight tutorials in the 4+ day "Using LLNL's Supercomputers" workshop. It is intended for those who are new to TotalView. A basic understanding of parallel programming in C or Fortran is required. The material covered in the following tutorials would also be beneficial for those who are unfamiliar with parallel programming in MPI, OpenMP and/or POSIX threads:

4.9 Valgrind

Visit

`http://valgrind.org/`

for more information. Here is an extract taken from that site.

Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. You can also use Valgrind to build new tools.

The Valgrind distribution currently includes six production-quality tools: a memory error detector, two thread error detectors, a cache and branch-prediction profiler, a call-graph generating cache and branch-prediction profiler, and a heap profiler. It also includes three experimental tools: a heap/stack/global array overrun detector, a second heap profiler that examines how heap blocks are used, and a SimPoint basic block vector generator. It runs on the following platforms:

X86/Linux,

AMD64/Linux,

ARM/Linux,

PPC32/Linux,

PPC64/Linux,

S390X/Linux,

MIPS/Linux,

ARM/Android (2.3.x and later),

X86/Android (4.0 and later),

X86/Darwin and

AMD64/Darwin

Mac OS X 10.6 and 10.7, with limited support for 10.8

Valgrind is Open Source and or Free Software, and is freely available under the GNU General Public License, version 2.

Here is an extract from the Wikipedia entry on Valgrind.

Valgrind is a GPL licensed programming tool for memory debugging, memory leak detection, and profiling. It is named after the main entrance to Valhalla in Norse mythology. Valgrind was originally designed to be a free memory debugging tool for Linux on x86, but has since evolved to become a generic framework for creating dynamic analysis tools such as checkers and profilers. It is used by a number of Linux-based projects.[3] Since version 3.5, Valgrind also works on Mac OS X. The original author of Valgrind is Julian Seward, who in 2006 won a Google-O'Reilly Open Source Award for his work on Valgrind. Several others have also made significant contributions, including Cerion Armour-Brown, Jeremy Fitzhardinge, Tom Hughes, Nicholas Nethercote, Paul Mackerras, Dirk Mueller, Bart Van Assche, Josef Weidendorfer and Robert Walsh. Valgrind is in essence a virtual machine using just-in-time (JIT) compilation techniques, including dynamic recompilation. Nothing from the original program ever gets run directly on the host processor. Instead, Valgrind first translates the program into a temporary, simpler form called Intermediate Representation (IR), which is a processor-neutral, SSA-based form. After the conversion, a tool (see below) is free to do whatever transformations it would like on the IR, before Valgrind translates the IR back into machine code and lets the host processor run it. Even though it could use dynamic translation (that is, the host and target processors are from different architectures), it doesn't. Valgrind recompiles binary code to run on host and target (or simulated) CPUs of the same architecture.

4.10 WinDbg

Van

I had been using totalview. With Lahey and Fujitsu 6.20e on Linux, it was the only debugger that understood their defective debugging tables – including the Fujitsu debugger.

But... the license for totalview is expensive. So when we switched from LF95 to Intel, and Intel provided idb, we started using idb.

I wish we still had totalview. idb is difficult to use, especially to view arrays. It's also rather slow.

Bill

A related topic would be corresponding compiler support. Typically compilers will optimize away parts of the code and discard symbol information, making the use of debuggers problematic. An option like "-g" often cures this, but at the expense of turning off most optimization. Options for intermediate cases between -g and (none) would be useful to document.

1) Print *

2) Totalview -

`www.roguewave.com/products/totalview.aspx`

3) DDT -

`www.allinea.com/products/ddt`

4) gdb and lgdb (parallel version of gdb)

Totalview, DDT, and lgdb support parallel codes, which is increasingly a requirement. gdb is free and powerful; the user interface has a 'high nerd factor'.

Chapter 5

Fortran aware editors and IDEs

Version 2.00, May 2012; Added an entry for Microsoft Visual Code

Version 1.9, June 2016; Added an entry for the Pathscale Dogfood IDE for Linux

Version 1.8, May 2016. Added an entry for Nag Fortran Builder for the Apple Mac and Windows operating systems.

Version 1.7, January 2015. Added Microsoft Visual Studio 2013 Community Edition.

Version 1.6, September 2014. Added Code::Blocks and geany entries.

Version 1.5, October 2011. Updated photran entry.

Version 1.4, July 2010. Added SunStudio Express entry.

Version 1.31, April 2006. Added Windows Zeus entry.

5.1 Windows

5.1.1 Absoft Editor (ae)

`http://www.absoft.com`

5.1.2 Code::Blocks

`http://www.codeblocks.org/`

5.1.3 CRiSP

`www.crisp.com` `http://www.crisp.demon.co.uk`

5.1.4 Compaq Visual Fortran 6.x

No longer available

5.1.5 editeur

`www.studioware.com`

5.1.6 emacs/xemacs - stand alone

`http://www.gnu.org/software/emacs/emacs.html`

5.1.7 emacs/xemacs - cygwin components

`http://www.cygwin.com/`

5.1.8 Fortran Builder for the Nag compiler

`http://www.nag.co.uk/nagware/NP/fortranbuilder.asp`

5.1.9 geany

`http://www.geany.org/`

5.1.10 gvim/vim - stand alone

`http://www.vim.org/`

5.1.11 gvim/vim - cygwin component

`http://www.cygwin.com/`

5.1.12 jed. wjed (Windows)

`http://www.jedsoft.org/jed/`

5.1.13 lahey ed

`http://www.lahey.com/`

5.1.14 Microsoft Visual Studio 6

No longer available.

5.1.15 Microsoft Visual Code

Microsoft make available a product called Visual Studio Code. Visit
`https://code.visualstudio.com/`
for more information.
Versions are available for Windows, Linux, and the macOS.
It can be configured to be Fortran aware.

5.1.16 Microsoft Visual Studio Community Edition

Microsoft have made Visual Studio available as a 'Community Edition'.

Visit

`https://visualstudio.microsoft.com/`

for more details. It is a free download. It includes Microsoft C++, C#, Visual Basic and F#. Here are details of some of the license conditions. This information has been taken from the Microsoft site.

> Any individual developer can use Visual Studio Community to create their own free or paid apps.
>
> Here's how Visual Studio Community can be used in organizations: An unlimited number of users within an organization can use Visual Studio Community for the following scenarios: in a classroom learning environment, for academic research, or for contributing to open source projects.

5.1.17 Microsoft Visual Studio.NET

There are a number of vendors that have provided integration with Microsoft Visual Studio .NET. These include

Intel Visual Fortran

`www.intel.com`

Lahey/Fujitsu Fortran

`www.lahey.com`

Silverfrost Salford FTN95

`http://www.silverfrost.com/11/ftn95/overview.aspx`

5.1.18 nedit - cygwin

`http://www.nedit.org/`

5.1.19 ntemacs

`http://www.gnu.org/software/emacs/windows/ntemacs.html`

5.1.20 photran

`http://www.eclipse.org/photran/`

Photran is an IDE and refactoring tool for Fortran based on Eclipse and the CDT. Photran is a component of the Eclipse Parallel Tools Platform (PTP).

5.1.21 Salford plato

`http://www.silverfrost.com/11/ftn95/overview.aspx`

5.1.22 UltraEdit

www.ultraedit.com

5.1.23 xemacs/emacs - stand alone

http://www.gnu.org/software/emacs/emacs.html

5.1.24 xemacs/emacs - cygwin components

http://www.cygwin.com/

5.1.25 Zeus ide

http://www.zeusedit.com/fortran.html

5.2 Linux/Unix

5.2.1 Code::Blocks

http://www.codeblocks.org/

5.2.2 CRiSP

www.crisp.com http://www.crisp.demon.co.uk

5.2.3 DF - Pathscale Dogfood

http://www.pathscale.com/DogFood

5.2.4 emacs/xemacs

http://www.gnu.org/software/emacs/emacs.html

5.2.5 geany

http://www.geany.org/

5.2.6 jed, xjed (Unix(all flavours)/OpenVMS) wjed (Windows)

http://www.jedsoft.org/jed/

5.2.7 Microsoft Visual Code

Microsoft make available a product called Visual Studio Code. Visit
`https://code.visualstudio.com/`
for more information.
Versions are available for Windows, Linux, and the macOS.
It can be configured to be Fortran aware.

5.2.8 nedit

`http://www.nedit.org/`

5.2.9 Oracle Solaris Studio

`http://developers.sun.com/sunstudio/downloads/express/`
Requires a Java run time.

5.2.10 photran

`http://www.eclipse.org/photran/`
Photran is an IDE and refactoring tool for Fortran based on Eclipse and the CDT. Photran is a component of the Eclipse Parallel Tools Platform (PTP).
Requires a Java run time.

5.3 Apple OS X

5.3.1 Absoft Editor

`http://www.absoft.com/`

5.3.2 BBEdit

`http://www.apple.com/downloads/macosx/productivity_tools/bbedit.html`

5.3.3 codeblocks

`http://www.codeblocks.org/`

5.3.4 emacs/xemacs

pre-installed

5.3.5 Fortran Builder for the Nag compiler

`http://www.nag.co.uk/nagware/NP/fortranbuilder.asp`

5.3.6 Microsoft Visual Code

Microsoft make available a product called Visual Studio Code. Visit
`https://code.visualstudio.com/`
for more information.
Versions are available for Windows, Linux, and the macOS.
It can be configured to be Fortran aware.

5.3.7 Photran

`www.photran.org/`

5.3.8 Smultron

`http://smultron.sourceforge.net/`

5.3.9 TextMate

`http://macromates.com/`

5.3.10 TextWrangler

`http://www.apple.com/downloads/macosx/productivity_tools/textwrangler.html`

5.3.11 Vim

Pre-installed

5.3.12 Xcode

`http://developer.apple.com/`

5.3.13 xemacs/emacs

pre-installed

Chapter 6

Commercial Courses

Version 1.7, October 2018. Added details of a new course from Sorcery Inc, Writing Fortran 2018 Today: Object-Oriented Parallel Programming.

Version 1.6, April 2016. Added a new NAG workshop.

Version 1.5, July 2015. Deleted John Reid entry. Corrected Mike Metcalf entry. Added Sourcery, Inc entry. Deleted Hector entry, added Archer entry. Updated the Cranfield entry.

Version 1.4, August 2012. Updated several entries.

Version 1.3, November 2011. Notified by Shaun Forth at Cranfield of their changes. Also added the HECToR entries.

Version 1.2, August 2010.

Version 1.1, September 2008

Version 1.0, January 2006.

6.1 Archer

Archer is the UK National Supercomputing Service.
`http://www.archer.ac.uk/`
The following list is taken from their web site.

```
Hands-on Introduction to High Performance Computing
Introduction to F95
Message-Passing Programming with MPI
Multicore Programming
Shared Memory Programming with OpenMP
Threaded Programming
Accelerator programming
Advanced MPI
Advanced OpenMP
```

Efficient Parallel IO on ARCHER
Performance Programming
PGAS Programming
Programming the Xeon Phi

6.2 Ian Chivers and Jane Sleightholme

Ian Chivers and Jane Sleightholme are available to do tailored on site courses. Courses include

Introduction to Modern Programming in Fortran

Advanced Features of Modern Fortran

See
http://www.fortranplus.co.uk

6.3 Cranfield University

Cranfield University, in conjunction with ISO Fortran Convener John Reid, offers two Fortran programming courses to groups of 6 or more when commissioned by an organisation:

Introduction to Programming in Fortran 2003 (3 days)

https://www.cranfield.ac.uk/courses/training/fortran-introduction-to-programming-in-fortran-2003.html

Intermediate Programming in Fortran 2003 (3 days) -

https://www.cranfield.ac.uk/courses/training/fortran-intermediate-programming-in-fortran-2003.html

These may be taken at the University's Shrivenham Campus or may be delivered at an organisation's site.

6.4 The Fortran Company

http://www.fortran.com/
Follow training links.

6.5 Lahey

http://www.lahey.com/
The Fortran 95 Workshop is a six-session, hands-on, Fortran 95 workshop led by Thomas M. Lahey, CEO, Lahey Computer Systems, Inc.

6.6 Michael Metcalf

formerly of CERN, Switzerland, and an ex-member of J3 and WG5, offers a Fortran 95 course that lasts for six 75-minute sessions. There is an F version too. He is happy to negotiate holding either version anywhere in the world. These courses are suitable for graduates, or equivalent level, and are a useful way kick-start a Fortran 90/95 or an F activity at a given site.

Contact

`michael.metcalf@t-online.de`

or

Manfred von Richthofen
Straße 15,
Berlin,
Germany,
+0049.30.78952573.

6.7 Nihon NAG, Numerical Algorithms Group Japan

Offers a Fortran Introduction course and Fortran consultancy. Their top page is

`http://www.nag-j.co.jp/`

They also have online material for their Fortran Introduction course, starting from

`http://www.nag-j.co.jp/fortran/index.html`

NAG also offer a two day workshop. Here is a short description.

> This two day computational science-focused practical hands on workshop is aimed at Fortran programmers who want to write modern code, or modernise existing codes, to make it more readable and maintainable by encouraging good software engineering practices. Adopting good software practices makes codes more amenable to optimisation and parallelisation, and the path to making it a community code a whole lot easier.

Topics will include:

> Software engineering for computational science;
>
> Modern Fortran standards and how to write optimised and efficient Fortran;
>
> NetCDF and HDF5 file formats for data sharing;
>
> GNU Automake to automate the build process;
>
> pFUnit unit testing framework;
>
> Doxygen for code documentation;
>
> Git version control for collaborative code development;

In-memory visualisation using PLplot;

Fortran interoperability with C, Python and R;

Introduction to parallelism for Fortran.

You need to contact
`WadudMiah[wadud.miah@nag.co.uk]`
for further information.

6.8 PTR Associates

Currently offer two Fortran courses.
`https://www.ptr.co.uk/fortran-programming`
Introduction To Modern Fortran Programming
`https://www.ptr.co.uk/advanced-fortran`
Advanced Programming in Modern Fortran

6.9 Purple Sage Computing Solutions, Inc

is offering three Workshops to Fortran programmers: The Fortran Modernization, Optimization and Parallelization Workshop; The Parallelization for Fortran Programmers Workshop; and The fthreads Workshop. Contact dnagle@erols.com or
`http://users.erols.com/dnagle`
Also on offer is a one day workshop on the new features of Fortran 2000. See
`http://users.erols.com/dnagle/wsf2000.html`
for more details.

6.10 Sourcery, Inc

`http://www.sourceryinstitute.org/`
Offer a range of courses. The following is taken from their web site.
Sourcery Inc training courses cover a range of topics relating to legacy and modern Fortran code development for computational science.

```
Parallel programming in Modern Fortran
Parallel numerical libraries for modern Fortran.
Parallel performance tuning and analysis with TAU.
Mixed Fortran/C/C++ programming
Software development methods, including
  Agile development
  Test-driven development and unit testing
  Literate programming
  Programming by contract
  Object-Oriented Analysis, Design, and Programming
Best practices in Object-Oriented Design (patterns)
```

They announced the following course in September 2018.

Writing Fortran 2018 Today: Object-Oriented Parallel Programming A 4-day course taught by Dr. Damian Rouson, Sourcery Institute and Izaak Beekman, ParaTools, Inc.

November 5-8, 2018, Sourcery Institute, Oakland, California, USA

`Register at http://bit.ly/writing-fortran-2018`

Fee: 2750 (professionals), 1375(students, postdocs, and remote participants)

Topics:

1. Introductory parallel programming: coarrays, synchronization, error termination, and collective subroutines.

2. Object-Oriented Programming: writing class hierarchies.

3. Advanced parallel programming: events, teams, and failed images.

4. Object-Oriented Design (OOD): best practices (design patterns).

The course will comprise roughly equal proportions of lecture, live coding demonstrations, and hands-on exercises. Attendees will work on their own computers inside Linux virtual machines containing the latest GNU Fortran, C, and C++ compilers; the OpenCoarrays parallel runtime library, and other open-source tools supporting a modern software-development workflow: porting a modern software-development workflow:

Distributed source management with git and GitHub.

Cross-platform building with CMake.

Unit testing with CTest.

Automatic documentation generation with FORD.

Reverse-engineering of OOD diagrams with ForUML.

Parallel performance engineering with TAU.

Attendees will also gain exposure to several agile development practices, including test-driven development, continuous integration, and pair programming.

Attendees receive a paperback of Scientific Software Design: The Object-Oriented Way by Rouson, Xia, and Xu (Cambridge Univ. Press, 2011)

Attendees will work on their own computers inside the Sourcery Institute Lubuntu Linux virtual machine available for free in the Sourcery Institute Store

6.11 France

Simulog, attn.

```
Mr. E. Plestan,
1 rue James Joule,
F-78286
Guyancourt Cedex,
France

Tel: +33 1 30 12 27 80
fax: +33 1 30 12 27 27

info@simulog.fr
```

6.12 Japan

6.12.1 Nihon NAG, Numerical Algorithms Group Japan

Offers a Fortran Introduction course and Fortran consultancy. Their top page is

`http://www.nag-j.co.jp/`

They also have online material for their Fortran Introduction course, starting from

`http://www.nag-j.co.jp/fortran/index.html`

A Japanese company offering courses and conversion consultancy is SofTek Systems, Inc. (see above).

Chapter 7

On Line Training Material

Version 1.1 August 2012. Removed Edinburgh entry - no longer available. Removed Manchester entry - no longer available.

Version 1.0 January 2006

7.1 CERN

`http://wwwinfo.cern.ch/asdoc/f90.html`

7.2 Paul Dubois

`http://prdownloads.sourceforge.net/pyfortran/OBF90.zip`.
lecture notes and class materials on Object Based Programming in Fortran 90 (In WinZip, on the Options—Configuration menu, turn off tar smart convert CR/LF.)

7.3 Linkoping University

`http://www.nsc.liu.se/~boein/f77to90/f77to90.html`
Fortran 77 to 90 Converson Course

7.4 Liverpool University

`http://www.liv.ac.uk/HPC/HPCpage.html`
Covers f90 and HPF, with Java-enhanced Web pages.

7.5 French

Support de cours Fortran 90 IDRIS - Corde & Delouis
`www.idris.fr/data/cours/lang/fortran/choix_doc.html`

Chapter 8

Graphics and Windows Programming

Version 1.5, May 2021: added an entry for ogpf, an object oriented interface to gnu plot. Added an entry for dolfyn, which is openGL based.

Version 1.4, April 2016, added GTK+, PLplot

Version 1.3, January 2013, added pgplot

Version 1.2, August 2012, updated various entries

Version 1.1, June 2009; updated web links.

Version 1.0, January 2006.

8.1 Introduction

This can be broken down into

Simple graphics programming using a library

visual interface via raw windows programming

visual interface via visual development environment

Here are some of the library and development offerings.

8.2 dislin

DISLIN is a high-level plotting library for displaying data as curves, polar plots, bar graphs, pie charts, 3D-color plots, surfaces, contours and maps.

`http://www.dislin.de/`
`http://www.mps.mpg.de/dislin/`
`http://www.mps.mpg.de/dislin/contents.html`

8.2.1 Worked examples

http://www.mps.mpg.de/dislin/examples.html

8.3 dolfyn

Here is an email we received regarding an OpenGL offering.

```
From: Henk Krus <h.krus@cyclone.nl>
Date: 29 April 2021 at 17:35:43 BST
To: queries@fortranplus.co.uk
Subject: Fortran information & resources - Graphics Libraries
Reply-To: h.krus@cyclone.nl

Dear Ian and Jane,

Recently I stumbled upon your Fortran Resources.
Referring to OpenGL and Worked Examples,
I might have a nice pointer for you:

https://dolfyn.net/dolfyn/f03gl_en.html

And the download in:

https://launchpad.net/f03gl

or the tarball in:

http://launchpad.net/f03gl/trunk/1.0/+download/f03gl_1.0.tgz

I worked through the various classic OpenGL books some years ago.
Played a bit a while ago with GTK+ as well. See

https://launchpad.net/f03gtk

One word of warning. Due to some developments at my ISP
(the Dutch equivalent of BT), the dolfyn.net site
will have to move to a new ISP before October this year.
My objective is to maintain, or redirect the main links.

By the way, the dolfyn CFD code is written in f03/f08 too.
You can find the code at

https://launchpad.net/dolfyn-cfd
```

```
I use the dolfyn code for my own work. The largest British
manufacturer of turbofan aircraft engines
used dolfyn as their base for combustion modelling.

Best regards,
Henk Krüs

PS: Currently also living with my Jane in Greenwich, London.
--
Cyclone Fluid Dynamics BV / Sweelincklaan 4 / NL-5583XM Waalre
T +31-(0)40-22 30 491 / www.cyclone.nl / www.dolfyn.net
```

8.4 gino

GINO is a suite of high-end development tools for creating complex 2D and 3D graphics and GUI applications. The products are ideally suited for aerospace, defense, utilities and other leading engineering organizations. The GINO products are available for Fortran, C/C++, VB, Delphi and .NET programming environments.

`http://www.gino-graphics.com`
`http://www.polyhedron.co.uk/`
`http://www.polyhedron.com/gino-ginomain0html`

8.4.1 Documentation

The software is supplied with on-line manuals in a variety of formats depending on the environment it is running (Windows Help, HTMLHelp, HTMLHelp2, PDF) and Printed Manuals are available at an additional cost.

`http://www.gino-graphics.com/downloads/manuals.htm`

8.4.2 Worked examples

None

8.5 ginomenu

GINOMENU is a subroutine toolkit for developing GUI applications under Windows. It provides extensive window and widget building modules allowing professional user-interfaces to be created under Windows 9x/NT/2000/XP without the need to get involved in MFC, API or mixed-language programming.

`http://www.gino-graphics.com/support.html`
`http://www.polyhedron.co.uk/`
`http://www.polyhedron.com/gino-ginomain0html`

8.5.1 Documentation

Windows HTML Help, PDF and printed documentation
`http://www.gino-graphics.com/support.html`

8.6 GTK+

`http://www.gtk.org/`

The following is taken from their site.

GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off tools to complete application suites

There are only partial Fortran bindings.

8.7 interacter

`http://www.polyhedron.co.uk/`

INTERACTER is our original multi-platform user-interface and graphics subroutine library for Fortran 77/9x developers.

8.7.1 Documentation

None

8.7.2 Worked examples

None

8.8 ogpf

`https://github.com/kookma/ogpf`

Here is an extract of a post made to
`comp.lang.fortran`

> The ogpf is an object based interface to GnuPlot from Fortran 2003, 2008 and later.
>
> It is a single file very clean and has a user friendly interface for direct plotting from Fortran!
>
> It has got an update to support FPM(fortran package manager)
>
> See the latest here:
>
> `https://github.com/kookma/ogpf`
>
> Best wishes
>
> Mohammad

Author:
Mohammad Rahmani
Chem Eng Dep., Amirkabir Uni. of Tech
Tehran, Ir
url: aut.ac.ir/m.rahmani

8.9 opengl

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

http://www.opengl.org/

8.9.1 Documentation

None

8.9.2 Worked examples

None at the above site.

Henk Krus has created some examples. Please see the earlier dolfyn entry for his openGL examples.

8.9.3 f90gl

f90gl is a public domain implementation of the official Fortran 90 bindings for OpenGL.

http://math.nist.gov/f90gl/

Precompiled f90gl libraries are available for some compilers. Lahey LF90, LF95 and ELF90:

http://www.lahey.com(searchforOpenGL)

Compaq CVF (formerly DVF):

http://www.compaq.com/fortran/

(click on "Downloads" and search for f90GL)

Intel Visual Fortran:

https://premier.intel.com/

(Registered users log in, select File Downloads and search for f90gl.)

Documentation

http://math.nist.gov/f90gl/documentation.html

Worked examples

Some precompiled libraries may not include the example programs or the source code for the examples. The following files contain the examples subdirectory from the f90gl distribution.

Unix: fglexamp.tar.gz gzipped tar file (73K) Win32: fglexamp.zip zip file (134K)

8.10 pgplot

`http://www.astro.caltech.edu/~tjp/pgplot/`

The following is taken from the pgplot site.

The PGPLOT Graphics Subroutine Library is a Fortran- or C-callable, device-independent graphics package for making simple scientific graphs. It is intended for making graphical images of publication quality with minimum effort on the part of the user. For most applications, the program can be device-independent, and the output can be directed to the appropriate device at run time.

The PGPLOT library consists of two major parts: a device-independent part and a set of device-dependent "device handler" subroutines for output on various terminals, image displays, dot-matrix printers, laser printers, and pen plotters. Common file formats supported include Postscript and GIF.

PGPLOT itself is written mostly in standard Fortran-77, with a few non-standard, system-dependent subroutines. PGPLOT subroutines can be called directly from a Fortran-77 or Fortran-90 program. A C binding library (cpgplot) and header file (cpgplot.h) are provided that allow PGPLOT to be called from a C or C++ program; the binding library handles conversion between C and Fortran argument-passing conventions.

PGPLOT has been tested with UNIX (most varieties, including Linux, SunOS, Solaris, HPUX, AIX, Irix, and MacOS X/Darwin) and OpenVMS operating systems. I am unable to provide support for DOS, Microsoft Windows, but I do distribute code provided by users for use with these operating systems.

8.11 PLplot

`http://plplot.sourceforge.net/`

The following is taken from the PLplot site.

PLplot is a cross-platform software package for creating scientific plots whose (UTF-8) plot symbols and text are limited in practice only by what Unicode-aware system fonts are installed on a user's computer. The PLplot software, which is primarily licensed under the LGPL, has a clean architecture that is organized as a core C library, separate language bindings for that library, and separate device drivers that are dynamically loaded by the core library which control how the plots are presented in noninteractive and interactive plotting contexts.

The PLplot core library can be used to create standard x-y plots, semi-log plots, log-log plots, contour plots, 3D surface plots, mesh plots, bar charts and pie charts. Multiple graphs (of the same or different sizes) may be placed on a single page, and multiple pages are allowed for those device formats that support them.

PLplot has core library support for plot symbols and text specified by the user in the UTF-8 encoding of Unicode. This means for our many Unicode-aware devices that plot symbols and text are only limited by the collection of glyphs normally available via installed system fonts. Furthermore, a large subset of our Unicode-aware devices also support complex text layout (CTL) languages such as Arabic, Hebrew, and Indic and Indic-derived CTL scripts such as Devanagari, Thai, Lao, and Tibetan. Thus, for these PLplot devices essentially any language that is supported by Unicode and installed system fonts can be used to label plots.

8.11.1 Documentation

`http://plplot.sourceforge.net/documentation.php`

8.11.2 Worked examples

`http://plplot.sourceforge.net/examples.php`

8.12 realwin

RealWin lets a Fortran programmer create full-featured applications for Microsoft 32-bit Windows platforms.

`http://www.indowsway.com/home.htm`
`http://www.indowsway.com/`

8.12.1 Documentation

Visit

`http://plplot.sourceforge.net/documentation.php`

8.12.2 Worked examples

Visit

`http://plplot.sourceforge.net/examples.php`

8.13 toolmaster

`http://www.avs.com/software/soft_t/toolm.html`

Toolmaster agX is a cross-platform graphics library. For FORTRAN programmers, AVS offers FGL/AGL, which provides equivalent functionality to the agX C library.

8.13.1 Documentation

None

8.13.2 Worked examples

http://www.avs.com/software/soft_t/toolm.html

8.14 winteracter

http://www.polyhedron.co.uk/

Winteracter is a modern GUI toolset for the Fortran 90/95 programming language. It consists of various visual development tools and a substantial subroutine library. Versions are available for most Fortran 9x compilers.

8.14.1 Documentation

None

8.14.2 Worked examples

None

8.15 Microsoft Windows graphics programming

This can be done in a variety of ways.

The following is a good book with examples of doing this using Compaq Visual Fortran.

Norman Lawrence, Compaq Visual Fortran: A Guide to Creating Windows Applications.

He also has coverage of opengl.

It is also possible to develop the visual interface using Visual Basic and call fortran dlls.

If you have Compaq Visual Fortran then the on-line Programmers Guide has coverage of mixed language programming with examples.

The following compilers offer integrated support for Windows programming under .NET.

8.15.1 Lahey/Fujitsu

http://www.lahey.com/

PRO for Windows adds a Fortran-smart Windows editor, a debugger, an AUTOMAKE make utility, and an enhanced Winteracter Starter kit (WiSK) for creating true Windows programs with Fortran, and a Coverage Analysis Tool that detects unexecuted code and performs range of operation checking. The PRO is compatible with Visual C++, Visual Basic, and Delphi and also includes Fujitsu's SSL2 Math Library and Visual Analyzer (see below).

8.15.2 Salford Software

`http://www.silverfrost.com/11/ftn95/overview.asp`

FTN95 for .NET, including integrated Help and Debugger, is supplied bundled with FTN95 for Win32 and, optionally, with Microsoft Visual Studio for .NET. A low-cost, fully-featured personal edition is also available.

Chapter 9

Parallel Programming

Version 2.2, September 2022. Minor updates, plus addition of Nvidia Cuda Fortran

Version 2.1, May 2021. Added an entry for Intel oneAPI.

Version 2.0, December 2019. Major rewrite to bring up to date with the current offerings. Intel MPI 2019 update 6 was released in December 2019.

Version 1.9, March 2015, added entry for the opencoarray project

Version 1.8, January 2013. Minor updates pointed out by Anton Shterenlikht

Version 1.7, January 2013. Added section on examples of combinations of compilers amd MPI libraries that work, or have worked in the past.

Version 1.6, August 2012; Updated several entries; Added several books;

Version 1.5, October 2011; Updated gfortran mpi entry; updated Intel coarray entry; updated NAG openmp entry

Version 1.4, August 2010; Updated coarray, mpi and openmp entries.

Version 1.3, July 2010; Added g95 entry. Modified gfortran entry.

Version 1.2, June 2009; Corrected and updated several web addresses.

Version 1.1, January 2006.

9.1 Introduction

The Fortran language has been standardised a number of times

Fortran 66

Fortran 77

Fortran 90

Fortran 95

Fortran 2003

Fortran 2008

Fortran 2018

The Fortran 90 standard added whole array features and a WHERE construct that were aimed at parallel programming.

The Fortran 95 standard added the FORALL construct, and PURE and ELEMENTAL procedures to help with parallel programming.

Fortran 2008 added coarrays.

The Fortran 2018 added additional coarray functionality.

Independently of the Fortran Standards Committees there have been a number of other developments aimed at parallel programming including

HPF

Intel oneAPI

MPI

Nvidia Cuda Fortran

OpenMP

Posix Threads

and each of these is covered in more depth below.

Two tutorials on parallel programming are given below.

`http://www.mhpcc.edu/training/workshop/parallel_intro/MAIN.html`

and

`http://users.actcom.co.il/~choo/lupg/tutorials/parallel-programming-theory/parallel-programming-theory.html`

9.1.1 Books

Rainer Keller (Editor), David Kramer (Editor), Jan-Philipp Weiss (Editor), Facing the Multicore-Challenge II: Aspects of New Paradigms and Technologies in Parallel Computing (Lecture Notes in Computer Science / Theoretical Computer Science and General Issues) Springer, 2012, ISBN-10: 364230396X, ISBN-13: 978-3642303968

K. De Bosschere (Author), E. H. D'Hollander (Author), G. R. Joubert (Author), D. Padua (Author), F. Peters (Author), Applications, Tools and Techniques on the Road to Exascale Computing, 2012, IOS Press, ISBN-10: 1614990409 ISBN-13: 978-1614990406

Kristjan Jonasson (Editor), Applied Parallel and Scientific Computing: 10th International Conference, PARA 2010, 2012, Springer, ISBN-10: 3642281508, ISBN-13: 978-3642281501

Victor Malyshkin (Editor), Parallel Computing Technologies: 11th International Conference, PaCT 2011, Kazan, Russia, September 19-23, 2011, Proceedings (Lecture Notes in ... Computer Science and General Issues), 2012, Springer, ISBN-10: 3642231772 ISBN-13: 978-3642231773

9.2 Automatic

By this is meant automatic parallelisation of the code without source code modification.

9.3 Coarray Fortran

Coarray Fortran is a small extension to Fortran 2003. It is a simple, explicit notation for data decomposition, such as that often used in message-passing models, expressed in a natural Fortran-like syntax. The syntax is architecture-independent and may be implemented not only on distributed memory machines but also on shared memory machines and even on clustered machines.

Coarray Fortran was the major component of the Fortran 2008 standard.

The Fortran 2018 standard added more functionality.

9.3.1 opencoarrays.org

The following

`http://www.opencoarrays.org/`

provides details of an open-source software project for developing, porting and tuning transport layers that support coarray Fortran compilers.

The Gfortran compiler from release 5 onwards is OpenCoarrays-compatible.

9.4 HPF

The High Performance Fortran Forum (HPFF), a coalition of industry, academic and laboratory representatives, works to define a set of extensions to Fortran 90 known collectively as High Performance Fortran (HPF). HPF extensions provide access to high-performance architecture features while maintaining portability across platforms.

Harvey Richardson has provided a historical perspective on HPF. Visit

`http://www.zeenty.com/HPF/HPF-intro.pdf`

Requires source code modification.

9.5 MPI

MPI is a library specification for message-passing, proposed as a standard by a broadly based committee of vendors, implementers, and users.

9.5.1 Openmpi

The Open MPI Project is an open source Message Passing Interface implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI is therefore able to combine the expertise, technologies, and resources from all across the High Performance Computing community in order to build the best MPI library available. Open MPI offers advantages for system and software vendors, application developers and computer science researchers.

Their home address is

`https://www.open-mpi.org/`

9.5.2 Mpich

MPICH is a high performance and widely portable implementation of the Message Passing Interface (MPI) standard.

MPICH and its derivatives form the most widely used implementations of MPI in the world. They are used exclusively on nine of the top 10 supercomputers (June 2016 ranking), including the world's fastest supercomputer: Taihu Light.

`https://www.mpich.org/`

`http://en.wikipedia.org/wiki/Message_Passing_Interface`

9.5.3 Intel MPI

Intel provide a free MPI library.

Visit `https://software.intel.com/en-us/mpi-library`
for more information.

9.5.4 Books

Aoyama, Yukiya; Nakano, Jun (1999) RS/6000 SP: Practical MPI Programming, ITSO. Available as a pdf.

`http://www.redbooks.ibm.com/abstracts/sg245380.html`

Gropp, William; Lusk, Ewing; Skjellum, Anthony (1999a). Using MPI, 2nd Edition: Portable Parallel Programming with the Message Passing Interface. Cambridge, MA, USA: MIT Press Scientific And Engineering Computation Series. ISBN 978-0-262-57132-6.

Pacheco, Peter S. (1997) Parallel Programming with MPI.[1] 500 pp. Morgan Kaufmann ISBN 1558603395.

Yiannis Cotronis (Editor), Anthony Danalis (Editor), Dimitris Nikolopoulos (Editor), Jack Dongarra (Editor) Recent Advances in the Message Passing Interface: 18th European MPI Users' Group Meeting, EuroMPI 2011, Santorini, Greece, September 18-21, 2011. ... / Programming and Software Engineering), 2011, Springer,ISBN-10: 3642244483, ISBN-13: 978-3642244483

9.5.5 Courses

In the UK the Hector service

`http://www.hector.ac.uk/`

provide various parallel programming courses. Details of their courses can be found at

`http://www.hector.ac.uk/cse/training/`

Cambridge University provide an MPI Course, offered by Nick Maclaren. See

`http://www-uxsup.csx.cam.ac.uk/courses/MPI/`

9.5.6 Requirements

Requires the installation of the MPI library (some compiler companies offer a bundle of compiler and MPI library) and source code modification.

9.6 oneAPI

Visit

`https://www.oneapi.com/`

for more information.

Here is some blurb from their site.

> oneAPI is a cross-industry, open, standards-based unified programming model that delivers a common developer experience across accelerator architectures,for faster application performance, more productivity, and greater innovation. The oneAPI industry initiative encourages collaboration on the oneAPI specification and compatible oneAPI implementations across the ecosystem.
>
> The oneAPI Specification
>
> The oneAPI specification extends existing developer programming models to enable a diverse set of hardware through language, a set of library APIs, and a low level hardware interface to support cross-architecture programming. To promote compatibility and enable developer productivity and innovation, the oneAPI specification builds upon industry standards and provides an open, cross-platform developer stack.
>
> The Language
>
> At the core of the oneAPI specification is DPC++, an open, cross-architecture language built upon the ISO C++ and Khronos SYCL standards. DPC++ extends these standards and provides explicit parallel constructs and offload interfaces to support a broad range of computing architectures and processors, including CPUs and accelerator architectures. Other languages and programming models can be supported on the oneAPI platform via the Accelerator Interface.
>
> The Libraries

oneAPI provides libraries for compute and data intensive domains. They include deep learning, scientific computing, video analytics, and media processing.

The Hardware Abstraction Layer

The low-level hardware interface defines a set of capabilities and services that allow a language runtime to utilize a hardware accelerator.

9.6.1 Intel oneAPI

Intel provide oneAPI toolkits that include a Fortran interface to parallelism using the oneAPI framework.

Visit

`https://software.intel.com/content/www/us/en/develop/tools/oneapi/all-toolkits.html#gs.zpx0jp`

OpenMP host and offload support is only available in the beta version of the Intel Fortran compiler - ifx.

9.6.2 Books

The following free book has coverage of a number of areas of general interest in parallel programming.

`https://www.apress.com/us/book/9781484255735`

Here is a bit of the blurb.

> Data Parallel C++ Mastering DPC++ for Programming of Heterogeneous Systems using C++ and SYCL Authors: Reinders, J., Ashbaugh, B., Brodman, J., Kinsner, M., Pennycook, J., Tian, X.

9.7 OpenMP

The OpenMP Application Program Interface (API) supports multi-platform shared-memory parallel programming in C/C++ and Fortran on all architectures, including Unix platforms and Windows NT platforms. Jointly defined by a group of major computer hardware and software vendors, OpenMP is a portable, scalable model that gives shared-memory parallel programmers a simple and flexible interface for developing parallel applications for platforms ranging from the desktop to the supercomputer.

`http://www.openmp.org/`

`http://en.wikipedia.org/wiki/OpenMP`

9.7.1 Books

> R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald, Parallel Programming in OpenMP. Morgan Kaufmann, 2000. ISBN 1558606718

> Parallel Programming in OpenMP, Chandra et al, 2007, Morgan Kaufmann, ISBN 978-1-55860-671-5

B. Chapman, G. Jost, R. van der Pas, D.J. Kuck (foreword), Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press (October 31, 2007). ISBN 0262533022

Using OpenMP, Chapman et al, 2007, MIT Press, ISBN 978-0262533027

Barbara Chapman (Editor), Federico Massaioli (Editor), Matthias S. Muller (Editor), Marco Rorro (Editor), OpenMP in a Heterogeneous World: 8th International Workshop on OpenMP, IWOMP 2012, Rome, Italy, June 11-13, 2012. Springer, ISBN-10: 3642309607 ISBN-13: 978-3642309601

Barbara M. Chapman (Editor), William D. Gropp (Editor), Kalyan Kumaran (Editor), Matthias S. Muller (Editor) , OpenMP in the Petascale Era: 7th International Workshop on OpenMP, IWOMP 2011, Chicago, Il, USA, June 13-15, 2011, Springer, ISBN-10: 364221486X ISBN-13: 978-3642214868

9.7.2 Courses

In the UK the Hector service
`http://www.hector.ac.uk/`
provide various parallel programming courses. Details of their courses can be found at
`http://www.hector.ac.uk/cse/training/`

9.7.3 Resources

`http://www.openmp.org/wp/resources/`
`http://www.openmp.org/wp/resources/openmp-compilers`
`http://openmp.org/wp/openmp-specifications/`

9.7.4 Requirements

Requires source code modification.

9.8 Nvidia Cuda Fortran

Visit the following site
`https://developer.nvidia.com/cuda-fortran`
for additional information.
Here are some extracts.

CUDA Fortran

CUDA Fortran is a software compiler and tool chain for building performance optimized GPU-accelerated Fortran applications targeting the NVIDIA GPU platform.

Powerful

CUDA Fortran includes language extension to simplify data management, the

```
!$cuf kernel
```

directive for automatically generating device code loops, and interface modules to all the NVIDIA CUDA-X math libraries.

Flexible

CUDA Fortran is designed to interoperate with other popular GPU programming models including CUDA C, OpenACC and OpenMP. You can directly access all the latest hardware and driver features including cooperative groups, Tensor Cores, managed memory, and direct to shared memory loads, and more.

Low Risk

CUDA Fortran is proven and supported on all major HPC platforms including x86-64, OpenPOWER and Arm-based servers. NVIDIA CUDA Fortran is available for use both on-premises and on all major cloud platforms including NGC. Commercial support options are available.

Key Features

Direct GPU Programming

CUDA Fortran is a low-level explicit programming model with substantial runtime library components that gives expert Fortran programmers direct control over all aspects of GPU programming. CUDA Fortran enables programmers to access and control all the newest GPU features including CUDA Managed Data, Cooperative Groups and Tensor Cores. CUDA Fortran includes several productivity enhancements such as Loop Kernel Directives, module interfaces to the NVIDIA GPU math libraries and OpenACC interoperability features.

The toolkit is currently only available for the linux platform.
Here is a link to their documentation.
`https://docs.nvidia.com/hpc-sdk/compilers/index.html`
Here is a link to their Cuda Fortran guide.
`https://docs.nvidia.com/hpc-sdk/compilers/cuda-fortran-prog-guide/index.html`

9.8.1 Books

Here is a link to some of books available.
`https://developer.nvidia.com/cuda-books-archive`

9.9 Posix Threads

Posix Threads is a library specification for multithreading, proposed as a standard by a broadly based committee of vendors, implementers, and users.
`http://www.llnl.gov/computing/tutorials/pthreads/`
Requires the installation of a threading library. Many operating systems come with a threading library pre-installed.

Also requires source code modification.
http://www.llnl.gov/computing/tutorials/pthreads/

9.10 Notes on the table below

Here is a quote from an email from Bill Long of Cray.
These interchanges took place on comp-fortran-90 at
http://www.jiscmail.ac.uk/lists/comp-fortran-90.html
The archives go back to 1997 and can be searched.

```
BEGIN QUOTE
>>
>>Erik Schnetter wrote:
>>
>>>
>>>Since MPI and threads are implemented as libraries, they
>>>work with
>>>every compiler.  They are on a rather low level.
>>>HPF and OpenMP are,
>>>in a way, language extensions that are translated into
>>>MPI or threads
>>>by the compiler.
>>>
>>>
>>>
>>Perhaps a bit simplistic to say
>>"they work with every compiler".
>>Whether MPI or threads (and what kind of threads) work
>>is generally
>>independent of the compiler, but not of the
>>operating system.   HPF,
>>OpenMP, and some forms of automatic parallelization
>>often involve
>>compiler generated calls to library routines, but
>>not necessarily to MPI or POSIX threads
>>library routines.
>>A vendor might opt for something more efficient.
>>
>>For many of the entries in Ian's list, there is an
>>implied combination of compiler, OS, and hardware.
>>For such a combination it is reasonable to talk
>>about support for MPI or pthreads.
>>Perhaps it would be helpful to be more explicit
>>about that combination.  Most of the parallel
>>programming schemes depend on more than
```

```
>>just the compiler.
>>
>>Cheers,
>>Bill
>>
END QUOTE
```

Here is a quote from an email from Malcolm Cohen of NAG

```
BEGIN QUOTE
>>
>>Erik Schnetter said:
>>> The IBM Fortran compiler supports Posix threads:
>>
>>As I suspect do most.  Certainly the NAG compiler does.
>>
>>Cheers,
>>--
>>.........................
>>Malcolm Cohen, Nihon NAG, Tokyo, Japan.
>>(malcolm@nag-j.co.jp)
>>
END QUOTE
```

In the light of these comments I've added a 'C' category which means that you will need to check your

hardware

operating system version

compiler version

MPI version or Posix Threads version

to see if the combination works.

The Y entry normally means that the compiler supplier provide a bundled or fully supported offering.

9.11 Table of compilers and supported parallel options

	Automatic	Co-Array Fortran	MPI	OpenMP	Posix Threads
Absoft	Y	N	C	Y	C
Arm		N	C	Y	C
Cray		Y	Y	Y	Y
Fujitsu		P	C	Y	C
gfortran		P	Y	Y	C
IBM		N	Y	Y	Y
Intel	Y	Y	Y	Y	C
NAG		Y	C	Y	Y
NEC		P	C	Y	C
Oracle		N	C	Y	C
PGI		N	C	Y	C

Parallelisation support

9.11.1 Coarray notes

Cray, Intel and Nag 7.0 offer full support.

Fujitsu, gfortran and Nec offer single image by default. The gfortran compiler can be linked with the opencoarrays library for full support.

9.12 Examples of setting up some Linux distributions for free MPI and coarray Fortran support with gfortran under Windows

In this section we show how to configure two Linux distributions for MPI and Coarray Fortran support. All compilers have openmp support built in.

Ubuntu 18.04.3 for can be configured to support parallel programming with MPI and Coarray Fortran.

cygwin can be configured for parallel programming support with MPI.

9.12.1 Ubuntu 18.04.3 and the Windows Subsystem for Linux

Follow the Microsoft documentation. Here are some links:

`https://docs.microsoft.com/en-us/windows/wsl/install-win10`

Once you have enabled the WSL you can use a web browser to visit the Microsoft site

`https://www.microsoft.com/en-gb/store/b/home`

and install Ubuntu 18.04.3. This version contains support for MPI by the openmpi project and coarrays using the opencoarrays project.

Once you have installed it start a session and use the lsb_release command.

Here is the output from running this command.

```
lsb_release -a

No LSB modules are available..
Distributor ID: Ubuntu
Decsription:    Ubuntu 18.04.3
Release:        Ubuntu 18.04
Codename:       bionic
```

You need to be running this version.

The following summarises the next step.

```
gfortran

Command 'gfortran' not found, but can be installed with

sudo apt install gfortran

mpif90

Command 'mpif90' not found, but can be installed with

sudo apt install libmpich-dev
sudo apt install libopenmpi-dev

caf

Command 'caf' not found, but can be installed with

sudo apt install open-coarrays-bin
```

Running

```
sudo apt install gfortran
sudo apt install open-coarrays-bin
sudo apt install libopenmpi-dev
```

creates a system that is capable of doing parallel programming in Fortran using openmp, mpi and coarray Fortran using the gfortran 7.4.0 release.

This install shares it file system with the host Windows operating system.

You have access to all of your files from the Ubuntu subsystem.

9.12.2 Ubuntu 18.04.3 and the Windows Hyper-V system

Follow the Microsoft documentation. Here are some links:

https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v

To do an install under Hyper-V we downloaded as ISO image of Ubuntu 18.04.3 and burnt a DVD.

We then followed the Hyper-V instructions to install a virtual machine (vm) for the Unbuntu installation.

We normally chose 8 GB of ram for the VM and 32GB of disk space for the Unbuntu virtual disk.

Start the machine and follow the instructions to install Ubuntu 18.04.3.

Follow the same instructions as before, i.e.

start a session and use the lsb_release command.

Here is the output from running this command.

```
lsb_release -a

No LSB modules are available..
Distributor ID: Ubuntu
Decsription:    Ubuntu 18.04.3
Release:        Ubuntu 18.04
Codename:       bionic
```

You need to be running this version.

The following summarises the next step.

```
gfortran

Command 'gfortran' not found, but can be installed with

sudo apt install gfortran

mpif90

Command 'mpif90' not found, but can be installed with

sudo apt install libmpich-dev
sudo apt install libopenmpi-dev

caf

Command 'caf' not found, but can be installed with

sudo apt install open-coarrays-bin
```

Running

```
sudo apt install gfortran
sudo apt install open-coarrays-bin
sudo apt install libopenmpi-dev
```

creates a system that is capable of doing parallel programming in Fortran using openmp, mpi and coarray Fortran using the gfortran 7.4.0 release.

This system has its own file system and is completely independent of the host Windows 10 system.

9.12.3 cygwin

The following has been taken from the cygwin site

`https://cygwin.com/index.html`

Cygwin is

> a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows.
>
> a DLL (cygwin1.dll) which provides substantial POSIX API functionality.

The first step is to download the setup program and run it.

Follow the instructions.

We recommend going for the basic install.

Next run the setup program again and choose [Category] view and type [gcc] into the search box.

Enable the installation of

> gcc core
>
> gfortran
>
> g++

Click [Next] and complete the install.

This has now set up cygwin to provide the ability to compile and run C, Fortran and C++ programs.

Run setup again, and choose [Category] view. Type [openmpi] into the search box and enable the installation of [openmpi].

Click next.

After completion you have a system that can compile and run openmp and MPI Fortran programs.

We recommend running setup again, choosing[Category] view and type [dos2unix] in the search box.

Install these utilities. This will allow you to convert between Dos and Unix text formats.

This system shares it file system with the host Windows system.

You have access to all of your files from the cygwin command prompt.

9.13 Parallelisation Tools

9.13.1 Crescent Bay Software

Offer a range of parallelisation tools.

`http://www.crescentbaysoftware.com/`

9.13.2 Parallel Software Products

Offer a tool to help parallelise Fortran 77, Fortran 90 or Fortran 95 code.

Their web address is

`http://www.parallelsp.com/index.htm`

The tool can generate either MPI calls or OpenMP calls.

Chapter 10

Fortran Software Tools

Version 1.7, October 2020. Updated the plusFORT entry and removed three others.

Version 1.6, April 2018. Updated the plusFORT entry.

Version 1.5, July 2015. Added Doctran entry. It is cross-platform documentation generation tool for the Fortran

Version 1.4. Updated the entry for Nag. The compiler can now produce call graphs, do a dependency analysis, produce interface modules and pretty print or polish Fortran code

Version 1.3, May 2013: Added Visustin Flow charting software. Updated various entries.

Version 1.2, 2011; Added refactoring as part of title; Add Photran entry;

Version 1.1, June 2009; Added web address for convert; Added web address for for_struct; Updated Nag entry - tools no longer available, being incorporated into the compiler.

Version 1.0, January 2006.

10.1 Refactoring

Wikipedia has a detailed coverage of code refactoring that is a good place to start. Have a look at

`http://en.wikipedia.org/wiki/Code_refactoring`

Here is their first paragraph.

Code refactoring is disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior, undertaken in order to improve some of the nonfunctional attributes of the software. Typically, this is done by applying series of refactorings, each of which is a (usually) tiny change in a computer program's source code that does not modify its functional requirements. Advantages include improved code readability and reduced complexity to improve the maintainability

of the source code, as well as a more expressive internal architecture or object model to improve extensibility.

Photran is listed in the Wikipedia entry as a Fortran refactoring tool.

10.2 Doctran

The following is taken from their web site.

Doctran is a cross-platform documentation generation tool for the Fortran programming language. It's purpose is to take free-format Fortran files, and produce a collection of linked html files documenting their contents.

Doctran can currently create documentation from source codes that contain constructs from the Fortran 95 standard and object oriented constructs from the Fortran 2003 standard.

`http://www.doctran.co.uk/`

10.3 Forcheck

A Fortran analyzer and programming aid, here is an extract describing the product.

> is the oldest and most comprehensive Fortran verifier on the market. It performs a full static analysis of an entire Fortran program or a separate analysis of one or more subprograms. detects more anomalies in your program than the compiler. Because it locates bugs as early in the development phase as possible, it saves you time and helps you to produce more reliable programs.
>
> is ideally suited to get a fast insight in existing and legacy programs. It composes optimal documentation with a call-tree and cross-reference tables both on the program-unit as on the program level.
>
> can be used as a software engineering tool in the various stages of the development process.
>
> can verify the conformance to the Fortran standards. Moreover it supports many language extensions of all popular compilers. FORCHECK is fully configurable so you can tune the analysis and output to your needs.
>
> can store the global information of the analyzed program-units in libraries. You can reference these libraries in subsequent FORCHECK runs to verify the consistency of all references and common-blocks.
>
> is very suitable as a cross-platform development tool.
>
> is available on many platforms from PC to supercomputer. It supports most Fortran extensions of all popular compilers.
>
> is sold with full guarantee and support.

`http://www.codework-solutions.com/development-tools/forcheck-fortran-analysis/`

10.4 Fortran90-lint

For Fortran 90 program analysis

`http://www.cleanscape.net/products/downloads/ftpflint.html`

10.5 NAG

The Nagware tools provided users with the ability to analyse and transform Fortran 77 and Fortran 95 code. They have been withdrawn as an individual product and the functionality is being added to the compiler.

The following options are currently supported:

=callgraph - Produce a callgraph of the Fortran routines in the source files.

=depend - Produce a dependency analysis of the Fortran source files.

=interfaces - Produce a module or include file containing procedure interfaces.

=polish - Pretty-print (polish) the Fortran source files.

`http://www.nag.co.uk/`

10.6 photran

Photran is an Integrated Development Environment (IDE) for Fortran 77, 90, 95, and 2003 based on Eclipse and the CDT. The project is maintained by the University of Illinois at Urbana-Champaign and IBM.

`http://www.eclipse.org/photran/`

10.7 plusFORT: Polyhedron Solutions

Their home address is:

`https://www.fortran.uk/`

Here is some information taken from their site.

> plusFORT, from Polyhedron Software, is a multi-purpose suite of tools for analyzing and improving Fortran programs. It combines restructuring and reformatting with global static analysis, dynamic analysis and many other features in a single powerful package. plusFORT is a one-stop solution for programmers, project managers, and quality assurance engineers working with Fortran source code.
>
> SPAG - Fortran source code restructuring, Quality Assurance using plusFORT, GXCHK - Global Static Analysis, Dynamic Analysis, Coverage Analysis.

10.8 Visustin

Automated flowcharting software. The following is taken from their web site.

Visustin is an automated flow chart program for software developers and document writers. Save documentation efforts with automatic code visualization. Visustin reverse engineers your source code to flow charts or UML Activity Diagrams. Visustin reads the if and else statements, loops and jumps and builds a diagram - fully automated.

No manual drawing is required. Visustin flowcharts ABAP, ActionScript, Ada, ASP, several assembly languages, BASIC, .bat files, C, C++, C Sharp, Clipper, COBOL, ColdFusion, Delphi, Fortran, Java, JavaScript, JCL (MVS), JavaServer Pages, LotusScript, Matlab, MXML, Pascal, Perl, PHP, PL/I, PL/SQL, PowerBuilder PowerScript, PureBasic, Python, QB, REALbasic, REXX, Ruby, SAS, Unix shell script (bash, csh, tcsh, ksh, sh), Tcl, TSQL, VB, VBA, VBScript, VB.Net, Visual FoxPro and XSLT.

`http://www.aivosto.com/visustin.html`

Chapter 11

Fortran Electronic Lists

Version 1.3, August 2012. Added Linkedin.

Version 1.2, November 2011. Expanded comp.lang.fortran entry with more information about usenet.

Version 1.1, January 2006.

11.1 comp-fortran-90

Jiscmail hosted. Restricted to questions about Fortran since the publication of the Fortran 90 standard. Can either browse on-line or subscribe and get postings via email. Postings are archived and go back to 1997.

`http://www.jiscmail.ac.uk/lists/comp-fortran-90.html`

11.2 comp.lang.fortran

Usenet news hosted Fortran list. Covers all aspects of Fortran.

Usenet is a worldwide distributed Internet discussion system. Users read and post messages (called articles or posts, and collectively termed news) to one or more categories, known as newsgroups. Discussions are generally threaded with modern news reader software.

Usenet is distributed amongst a large, changing set of servers that store and forward messages to one another in so-called news feeds. Individual users may read messages from and post messages to a local server operated by their Internet service provider, university, or employer.

Newsreader clients

Newsgroups are typically accessed with special client software that connects to a news server. Newsreader clients are available for all major operating systems.

Web accessible newsgroups

Web front ends to newsgroups mean that many people now no longer need to use download and install or configure a news reader client Google Groups is one such web based front end and web browsers can access Google Groups.

Free usenet news service

`http://www.eternal-september.org/`

Welcome to news.eternal-september.org news.eternal-september.org is a private project providing free access to text-only Usenet News. The server has a 100MBit connection to several Internet backbones and is integrated into the Usenet via more than 60 peers.

Free access to the news server news.eternal-september.org provides free read and write access to all text newsgroups. It requires a registration that can be done online.

`http://groups.google.co.uk/group/comp.lang.fortran?lnk=lr` `http://groups.google.ca/group/comp.lang.fortran`

11.3 Fortran Discourse

Their home page is:

`https://fortran-lang.discourse.group/`

Here is an extract from their site.

> Welcome to the Fortran Discourse!
>
> About This forum is for help, discussion, and announcements related to the Fortran Programming Language
>
> About Fortran Discourse A community of developers of Fortran language, libraries, and applications
>
> Our Admins
>
> milancurcic certik Ondřej Čertík
>
> Our Moderators
>
> lkedward Laurence Kedward

11.4 LinkedIn

LinkedIn is a social networking website for people in professional occupations. As of June 2012, LinkedIn reports more than 175 million registered users in more than 200 countries and territories.

`http://www.linkedin.com/`

The Fortran Programmers Group has 1697 members as of August 2012.

The site is available in English, French, German, Italian, Portuguese, Spanish, Dutch, Swedish, Romanian, Russian, Turkish, Japanese, Czech, Polish, Korean, Bahasa Indonesia, and Bahasa Malaysia.

11.5 Compiler specific

Some of the compiler suppliers provide electronic list support. Some provide an email address for technical support.

11.5.1 Absoft

`http://forums.absoft.com/`

`support@absoft.com`

11.5.2 Apogee

`info@apogee.com`

11.5.3 Compaq

11.5.4 Cray

11.5.5 Fortran Company

You can subscribe to an e-mail list by sending e-mail to majordomo@fortran.com with the following in the body of the message:

subscribe f-interest-group@fortran.com

11.5.6 Fujitsu

11.5.7 Gnu Fortran 95

You can reach us at the fortran@gcc.gnu.org mailing list; for details please refer to our mailing lists page.

`http://gcc.gnu.org/lists.html`

11.5.8 G95

`http://groups.google.com/group/gg95`

11.5.9 Hewlett Packard

`vf-support@hp.com`

11.5.10 IBM

Requires registration.

11.5.11 Intel

Requires registration.
`https://premier.intel.com/WhatsNew.aspx`

11.5.12 Lahey Fujitsu

Requires registration.
`http://www.lahey.com/support.htm`
`http://www.laheyforum.com/`

11.5.13 NAG

Requires registration.

`support@nag.co.uk`

11.5.14 NA Software

11.5.15 NEC

`http://www.nec.com/global/support/index.html`

11.5.16 Pathscale

Requires registration.
`http://pathscale.com/support.html`

`support@pathscale.com`

11.5.17 PGI

Various offerings.
`http://www.pgroup.com/support/index.htm`

11.5.18 Salford Software

Various options.
`http://www.silverfrost.com/22/ftn95/support/index.asp`

11.5.19 SGI

Various options.
`http://www.sgi.com/support/customerservice.html`

11.5.20 SUN

http://forums.sun.com/category.jspa?categoryID=113

Chapter 12

Fortran Standard Bodies

Version 1.2, April 2018. Major update to the WG5 and J3 entries to bring them up to date.

Version 1.1, August 2012; Updated J3 entry to reflect organisation within the US

Version 1.0, January 2006.

12.1 Introduction

There are two main Fortran standards bodies and these are WG5 and PL22.3 (formerly J3). Each is covered in turn below.

12.2 WG5

Their home page is:-
`https://wg5-fortran.org/`
Here are some details about the people involved.

WG5 Convenor: Steve Lionel (US)

ISO/IEC 1539-1 Editor: Malcolm Cohen (UK) (Base Language)

Corrigenda Editor: David Muxworthy (UK)

ISO/IEC 1539-2 Editor: John Reid (UK) (Varying Length Strings)

ISO/IEC TS 29113 Editor: Bill Long (USA) (Further Interoperability of Fortran with C)

ISO/IEC TS 18508 Editor: Bill Long (USA) (Additional Parallel Features in Fortran)

Here are details of the convenors and editors.

John Reid (UK) was WG5 Convenor from 1999 through 2017. From 1995 until 1999, the Convenor of WG5 was Miles Ellis (UK). His predecessor was Jeanne Martin (USA), who was Convenor from 1982 until 1994.

Prior to that Jeanne Adams (USA) was Convenor of WG5 and its predecessor, the Fortran Experts Group, from the creation of the latter in 1978 until 1982.

The editor of the Fortran 2003 Standard (ISO/IEC 1539-1:2004(E)) was Richard Maine (USA).

The editor of the Fortran 95 Standard (ISO/IEC 1539-1:1997) was Richard Maine (USA).

The editor of the Fortran 90 Standard (IS 1539:1991) was Lloyd Campbell (USA) during most of its development and Mike Metcalf (CERN) during the final stages.

12.3 PL22.3

Task Group PL22.3 (formerly J3) is responsible for the development and interpretation of the United State and International standards for Programming Language Fortran.

This technical committee is the U.S. TAG to ISO/IEC JTC1 SC22/WG5

Group participants (April 2018) include

```
Corbett
Cray Inc
Hendrickson
IBM Corporation
Intel Corporation
Jet Propulsion Laboratory
Kernelyze LLC
Lawrence Berkeley National Laboratory
Lionel
Maine
NASA
NVidia Corporation
National Center for Atmospheric Research (NCAR)
United States Dept of Energy
```

`http://standards.incits.org/a/public/group/pl22.3`

The PL22.3 - J3 home page is:-

`http://www.j3-fortran.org/`

Their list of members (April 2018) is given below.

```
Voting Principals
Bryce Adelstein-Lelbach
      Lawrence Berkeley National Laboratory
Daniel Chen
```

IBM Corp.
Thomas Clune
NASA GSFC
Robert Corbett
Robert Corbett (self)
Thomas Knox Kernelyze
LLC
Gary Klimowicz
Nvidia Corporation
Steve Lionel
Steve Lionel (Self) WG5 Convenor
William Long
Cray Inc.
Lorri Menard
Intel Corporation
Karla Morris
Sandia National Laboratories
Dan Nagle
Chair J3, National Center for Atmospheric Research
Craig E Rasmussen
Van Snyder
Jet Propulsion Laboratory
Voting alternatives
Malcolm Cohen
Craig Rasmussen
Brian Friesen
Bryce Adelstein-Lebach
Andrew Gontarek
William Long
Henry Jin
Tom Clune
Mark LeAir
Gary Klimowicz
Kelvin Li
Daniel Chen
Raghu Maddhipatla
Lorri Menard
Divya Mangudi
Lorri Menard
Toon Moene
Karla Morris
John K. Reid
Dan Nagle
Damian Rouson
Karla Morris
Dr Anton Shterenlikht

```
      Thomas Knox
Jon Steide
      Lorri Menard
Dr. John Wallin
      Dan Nagle
Rafik Zurob
      Daniel Chen
```

Visit
`https://j3-fortran.org/members.php`
for up to date information.
Their version of the working draft is at the J3 site.
`https://j3-fortran.org/doc/year/18/18-007.pdf`

12.4 ISO - International Organization for Standardization

Their home page is:
`https://www.iso.org/home.html`
Here is some blurb from their site.

> ISO creates documents that provide requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose.
>
> We've published 22677 International Standards, which you can buy from our members or the ISO Store.
>
> ISO is an independent, non-governmental international organization with a membership of 164 national standards bodies.
>
> Through its members, it brings together experts to share knowledge and develop voluntary, consensus-based, market relevant International Standards that support innovation and provide solutions to global challenges.
>
> International Standards make things work. They give world-class specifications for products, services and systems, to ensure quality, safety and efficiency.

WG 5 reports to the ISO.
Here is the hierarchy:

> ISO/IEC JTC 1 — Information Technology
>
> JTC 1 is the standards development environment where experts come together to develop worldwide Information and Communication Technology (ICT) standards for business and consumer applications.
>
> – ISO/IEC JTC 1/SC 22
>   Programming languages, their environments and system software interfaces

The following 4 publications are the current Fortran standards available from the ISO site.

`https://www.iso.org/standard/72320.html`

Fortran 2018.

`https://www.iso.org/standard/26934.html`

Varying length character strings.

`https://www.iso.org/standard/45136.html`

Further interoperability of Fortran with C.

`https://www.iso.org/standard/62702.html`

Additional Parallel Features in Fortran.

The status of some of these will be reviewed at the next standards meeting in Japan in August 2019.

Chapter 13

Fortran History

March 2020. Updated the Lorenzo entry. Mike Metcalf has done a review for Fortran Forum.

October 2019. Added the Lorenzo book.

August 2019. Updated to have the publication dates of all ISO Fortran standards.

version 1.2, June 2019. Added ISBN details for most of the book entries. Added a link to the British Library catalogue. The catalogue was used to determine the ISBN numbers where available.

Version 1.1, March 2019. Updated with additional material, especially what is available on line at the IEEE and ACM.

Version 1.0, January 2019. There is now a separate chapter on Fortran's history.

13.1 Fortran up to the first standard

13.1.1 John Backus, 1957

Early paper by Backus. A reprint appears in: Programming Systems and Languages (S. Rosen ed.), McGraw Hill, 1967, pp. 29-47. ISBN 10: 0070537089, ISBN 13: 9780070537088

13.1.2 Two early text books on Fortran II

A Guide to FORTRAN Programming, McCracken, Daniel D., Wiley, 1961. ISBN 10: 0471582123 ISBN-13: 978-0471582120

A FORTRAN Primer, Organick, E.I., Addison-Wesley, 1963. No ISBN is available for this book.

13.1.3 History of FORTRAN and FORTRAN II

Here is the abstract taken from the site.

> The goal of this project is to preserve source code, design documents, and other materials concerning the original IBM 704 FORTRAN/FORTRAN II compiler. FORTRAN was the first high-level programming language and the first high-quality optimizing compiler. This is a project of the Computer History Museum's Software Preservation Group to develop expertise in the collection, preservation, and presentation of historic software. Comments, suggestions, and donations of additional materials are greatly appreciated.

Here is a link to the site.

`http://www.softwarepreservation.org/projects/FORTRAN/`

13.1.4 A brief history of FORTRAN-Fortran

`http://www.ibiblio.org/pub/languages/fortran/ch1-1.html`

13.1.5 Annals of the History of Computing, Volume 6, Number 1, January - March, 1984

This edition is dedicated to the history of Fortran.

J. A. N. Lee and, Henry S. Tropp, About This Issue, 3–4

Anonymous, Contributors, 4–6

J. A. N. Lee, Pioneer Day, 1982, 7–14

John Backus, Early Days of FORTRAN, 15–15

John C. McPherson, Early Computers and Computing Institutions, 15–16

Robert W. Bemer, Computing Prior to FORTRAN, 16–18

Richard Goldberg, Register Allocation in FORTRAN I, 19–20

Roy Nutt, Compiler Techniques Available in 1954, 20–22

Frances E. Allen, A Technological Review of the Early FORTRAN Compilers, 22–25

John Backus, Afterword, 26–27

Jeanne Adams, Institutionalization of FORTRAN, 28–28

Herbert S. Bright, Early FORTRAN User Experience, 28–30

Robert A. Hughes, Early FORTRAN at Livermore, 30–30

William P. Heising, The Emergence of FORTRAN IV from FORTRAN II, 31–32

Martin N. Greenfield, The Impact of FORTRAN Standardization, 33–33

Daniel D. McCracken, The Early History of FORTRAN Publications 33–34

Charles Davidson, The Emergence of Load-and-Go Systems for FORTRAN 35–37

James M. Sakoda, A Dynamic Storage Allocation Language - DYSTAL, 37–38

Bruce Rosenblatt, The Successors to FORTRAN - Why Does FORTRAN Survive? 39–40

Daniel N. Leeson, IBM FORTRAN Exhibit and Film, 41–48

J. A. N. Lee, An Annotated Bibliography of FORTRAN, 49–58

Henry S. Tropp, FORTRAN Anecdotes, 59–64

Elliott C. Nohr, Meetings in Retrospect; FORTRAN Celebration at IBM Santa Teresa Laboratory; FORTRAN Activities at SHARE 59 Meeting, 65–69

Anonymous, Self-Study Questions and Answers, 69, 72–73

Anonymous, News and Notices. 70–71

K. W. Smillie, Reviews: FORTRAN Papers from NCC 82 Proceedings; Capsule Reviews 74–80

At the following IEEE site

`https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=4392944`

The following documents are available.

About this issue, Page(s):3 - 4

Contributors, Page(s):4 - 6

Pioneer Day, 1982, J.A.N. Lee, Page(s):7 - 14

Early Computers and Computing Institutions, John C. McPherson, Page(s):15 - 16

Early FORTRAN User Experience, Herbert S. Bright, Page(s):28 - 30

IBM FORTRAN Exhibit and Film, Daniel N. Leeson, Page(s):41 - 48

An Annotated Bibliography of FORTRAN, J.A.N. Lee, Page(s):49 - 58

FORTRAN Anecdotes, Page(s):59 - 64

Meetings in Retrospect, Page(s):65 - 69

The McPherson, Bright and Leeson contributions are only available to IEEE members. The rest are available as a free pdf download.

13.1.6 The History of FORTRAN I, II, and III, John Backus

In ACM Sigplan History of Programming Languages Conference - Preprints, published in ACM Sigplan Notices 13(8), pp. 165-180, August 1978.

13.1.7 History of Programming Languages, R.L. Wexelblat ed., Academic Press, 1981, pp. 25-74, ISBN 0-12-745040-8

The book is in the History of Programming Language Series as an ACM Monograph.

Here is an extract from the book.

> These proceedings of the ACM SIGPLAN History of Programming Languages (HOPL) conference are a record, in the words of those who helped make the history, of a baker's dozen of the languages that set the tone for most of today's programming.

The speaker on Fortran for the second session of the conference was John Backus. His paper is entitled The History of FORTRAN I, II and III

The Backus paper has coverage of

Early background and environment

The early stages of the Fortran project

The construction of the Compiler

FORTRAN II

FORTRAN III

FORTRAN after 1958: Comments

References

The session also has details of

Transcript of presentation

Transcript of discussants remarks

Transcript of Q and A session

Full text of all questions submitted

Visit

`https://dl.acm.org/citation.cfm?id=800025`

for information about the on line version.

The paper is available for 15 to download in PDF format for non-members.

13.1.8 ACM Sigplan Notices 13(8), pp. 165-180, August 1978

The previous publication also appears under the ACM Sigplan banner.

Visit

`https://dl.acm.org/citation.cfm?id=960118`

for more information.

13.1.9 Abstracting Away the Machine: The History of the FORTRAN Programming Language (FORmula TRANslation)

Recent (2019) book on the history of Fortran. The author is Mark Jones Lorenzo and the publisher is Amazon Digital Services LLC - KDP Print US, 2019. ISBN: 1082395943, 9781082395949. Length 326 pages.

Here is some blurb from the Amazon site.

> At the dawn of the computer age, an elite development team at IBM built the most influential computer programming language in history: FORTRAN. Abstracting Away the Machine tells the epic story of how they did it–and what happened next.

Mike Metcalf has done a review in the April edition of Fortran Forum. In his words

> It becomes the granddaddy of Fortran history sources.

Worth a read.

13.2 The run up to the Fortran 90 standard

13.2.1 The Fortran (not the foresight) saga: the light and the dark

An article written by Brian Meek in 1990 on the development of Fortran 90. A version can be found at `https://www.fortranplus.co.uk/fortran-information/`

and the title is a play of words based on

`https://en.wikipedia.org/wiki/The_Forsyte_Saga`

which was a book and British TV series.

13.2.2 A Personal History of the NAG Fortran compiler, Malcolm Cohen, Tokyo, October 2004

Pre History

The year was 1988. The revision of the Fortran standard had fallen into turmoil.

The committee which was revising the language standard for ISO, X3J3 (the U.S. Fortran committee, which has since changed its name to J3), was split roughly into three warring factions:

1 The progressives: those who wanted to very greatly modernise the language (mostly users).

2 The traditionalists: those who wanted to freeze the language, or perhaps just adopt extremely minor extensions (mostly vendors).

3 The moderates: those who wanted to modernise the language, but not by as much as the first group (some users, some vendors).

After the end of the public consultation period in 1987, these three groups could not agree on how to proceed. The traditionalists were saying that the proposed new language could not be implemented. Each moderate was saying that the proposed language was too complicated, and besides was missing his favourite feature (a different one for each person). The progressives were saying that the others just wanted to ruin the new language by taking away the good bits. (Note: In reality it was a good deal more complicated than this, but this is a reasonable simplification which gives some idea of the problem at the time.)

The issue came to a head in the middle of 1988, at the meeting in Paris of the ISO working group on Fortran (this committee rejoices in the name of ISO/IEC JTC1/SC22/WG5). This working group has the responsibility for revising the Fortran standard, though then as now the technical work was essentially delegated to the American committee. Due to the impasse on the American committee, WG5 decided that it was its responsibility to decide what should be in the new language, and therefore discussed the competing proposals.

In response to the traditionalists' claim that the language was so complex and unwieldy that it could never be implemented, at this meeting Julian Tilbury of Salford Software and myself presented a front-end for Fortran 8x (the working name for the language revision) suitable for building software tools or indeed a compiler. It was significant that this demonstration (which analysed the complete, supposedly non-implementable, language) had taken only 3 months to write from scratch. (The demonstration did not quite go without a hitch, because the keyboard was French but the software treated it as English. So when typing in examples I had to close my eyes - or stare fixedly at the screen - and type where I remembered the keys ought to be instead of looking at the keyboard.)

In the event, WG5 decided on a relatively progressive modernisation of the language which would address the concerns expressed by the international community, and specified what new features the language should contain. In more hope than expectation that the work could be quickly done WG5 suggested that the new language be called Fortran 88, and this then became Fortran 90 after the work took longer than hoped.

NAG has built its reputation on a number of fine products, but by far the most important in terms of customers was its numerical library in Fortran. It was clear to NAG that Fortran continued to offer the best facilities for writing numerical software, and that NAG had an interest in Fortran maintaining its market share. It was also believed that Fortran would simply wither away if it were not quickly modernised, which is why NAG supported the efforts of the standards committees by attending these meetings (and also in helping to produce the demonstration in Paris '88).

NAG needed to have early access to Fortran 90 compilers, so that it could develop its products ready for when the compilers became widely available. Furthermore, having made extensive successful use of Fortran (66, then 77) software tools in the development and quality assurance of its Fortran (66/77) library, it wanted also to have access to Fortran 90 software tools. It was also thought that NAG producing a compiler would act as a spur to the other compiler vendors, and also ensure that even if a native vendor compiler was not available for some particular machine, we would have our own that we could use.

It was in the late spring of 1990, in conversation with one of the directors of NAG about the future of Fortran, that it was suggested that I might be able to write a compiler for the new language. I replied that naturally I thought I could do such a thing, having had a long-standing interest in programming language design and implementation, and indeed had designed a language of my own and written a compiler for it while I was at college (a much smaller language than Fortran 90, though!).

Perhaps I should not have been surprised at the turn of events ... first my manager asked me how long I thought it would take me to write a compiler (I answered "definitely more than a year; say about fifteen months?"). The next thing I knew was that had become my assigned task and I had a deadline set at fifteen months! Needless to say, during that development period I worked harder and longer than ever before on a single product.

The first decisions to be taken were the target machine or language, and the implementation language. I decided both to write the compiler in C, and to produce C (as a portable assembler). Some of the reasons being

> I was sufficiently familiar with C to be able to write highly portable code.
>
> C compilers were widely available (if less so than now).
>
> It would allow the compiler to be very quickly ported to new systems.
>
> I could use the C compiler optimiser, reducing the amount of optimisation that the Fortran compiler needed to do itself.

Most of the Fortran 90 features were familiar to me, as I had encountered similar features in other programming languages. One problem was simply the number of features and their interactions. It is possible to put on an extra burst of energy to reach the top of a hill that is within sight, but maintaining that level of output when climbing a mountain is not so easy. One gets to the top of one peak, only to be greeted with the sight of additional vistas opening before you.

The largest feature that I was relatively unfamiliar with was the array syntax and the whole array operations. Since for the first release in particular, correctness was a much

more important goal than performance, I took a very simple approach to handling array expressions. Since I was unsure exactly how to evaluate a whole array expression at once, but knew how to evaluate any single array operation, every array expression was broken down into single operations producing a temporary array result. This approach was very successful in guaranteeing the right answer, but unfortunately also quite successful in not providing good performance. (This is why, in release 2.0 a couple of years later, the array expression handling was completely rewritten.)

Another major problem was that of testing the compiler. For the subset that was in Fortran 77 there were existing test suites we could (and did) use. For numerical accuracy we had our own tests and library. But for all the new features there was, of course, virtually no code in existence - just a few examples in the standard itself and in books such as Mike Metcalf and John Reid's "Fortran 90 Explained" (we found this book so clear in its explanations of the new language that we decided to use it as the compiler manual).

I wrote many individual feature test programs myself during development, but this was not sufficient to ensure that features would work when used together. We knew that Brian Smith and others were writing a test suite, so in exchange for use of the prototype compiler to test their test programs, we got bug reports back when the compiler went wrong. This proved to be invaluable, and doubtless without this it would have taken many months longer to bring the compiler to a releasable quality.

By the spring of 1991 I found that I needed a break from doing the compiler all day, so in the evenings I decided to build a car (a Caterham Super Seven). It seems that cars must be simpler than compilers though, as I finished the (high performance) car first!

And so, after many trials and tribulations, the compiler was finally released (as the world's first Fortran 90 compiler) in September 1991, at virtually the same time that the standard saw publication.

For all my hard work in producing the compiler I was rewarded with the tasks not only of supporting and maintaining it, but to extend and enhance it to encompass new language standards (HPF, then Fortran 95 and now Fortran 2003), user requests for extensions (many, mostly historical "dusty-deck" style language), greater performance, and even more facilities for detecting errors in programs both at compile time and execution time. As I write, the current release of the compiler already includes a substantial number of Fortran 2003 features, and work is underway on further improvements.

Additional history of the Nag compiler

The following is taken from an interchange with Malcolm Cohen of Nag.

> Yes there was DEC VAX/VMS, and later DEC Alpha OpenVMS. We also did DEC Alpha Unix.
>
> The VMS support stopped before 2002, but release 4.2 still supported DEC Alpha Unix, DEC Alpha Linux, HP 9000/700, IBM RS6000, SGI Irix 5 and 6, Itanium Linux, Sun 4.0 and 5.x, IBM OS/390 Open Edition, Linux x86, FreeBSD x86, and very basic support for Windows (no gui or debugger, command line only).

13.3 ISO Fortran standardisation history

The following table has been created from information available at the ISO site. It documents the base Fortran standard history.

ISO Reference	Corrigenda Amendments	Year	Month	Name
1539:1972		1972		Fortran 66
1539:1980		1980	March	Fortran 77
1539:1991		1991	July	Fortran 90
	1539:1991/Cor 1:1994	1994	August	
	1539:1991/Cor 2:1995	1995	June	
	1539:1991/Cor 3:1997	1997	July	
1539-1:1997		1997	December	Fortran 95
	1539-1:1997/Cor 1:2001	2001	June	
	1539-1:1997/Cor 2:2002	2002	June	
1539-1:2004		2004	November	Fortran 2003
	1539-1:2004/Cor 1:2006	2006	February	
	1539-1:2004/Cor 2:2007	2007	February	
	1539-1:2004/Cor 3:2008	2008	November	
	1539-1:2004/Cor 4:2009	2009	September	
1539-1:2010		2010	October	Fortran 2008
	1539-1:2010/Cor 1:2012	2012	October	
	1539-1:2010/Cor 2:2013	2013	June	
	1539-1:2010/Cor 3:2014	2014	September	
	1539-1:2010/Cor 4:2016	2016	July	
1539-1:2018		2018	November	Fortran 2018

Fortran standardisation history

All but the last have been withdrawn.

The following table documents the TR and TS standard history.

ISO Reference	Year	Month	Name	Status
1539-2:1994	1994	December	Varying length character strings	Withdrawn
1539-2:2000	2000	June	Varying length character strings	Current
1539-3:1999	1999	February	Conditional compilation	Withdrawn
TR 15580:1998	1998	December	Floating-point exception handling	Withdrawn
TR 15580:2001	2001	June		Withdrawn
TR 15581:1998	1998	December	Enhanced data type facilities	Withdrawn
TR 15581:2001	2001	June		Withdrawn
TR 19767:2005	2005	February	Enhanced module facilities	Withdrawn
TS 29113:2012	2012	December	Further interoperability of Fortran with C	Current
TS 18508:2015	2015	December	Additional Parallel Features in Fortran	Current

Fortran TR and TS standardisation history

13.4 Miscellaneous links

13.4.1 The Seven Ages of Fortran

Can be found at

`http://journal.info.unlp.edu.ar/journal/journal30/papers/JCST-Apr11-1.pdf`

13.4.2 Computer Languages History (preview)

`http://www.levenez.com/lang/history.html`

13.4.3 Computer Languages History

`http://www.levenez.com/lang/`

13.4.4 Open Directory - Fortran Tutorials Fortran 90 and 95

`http://www.dmoz.org/Computers/Programming/Languages/Fortran/Tutorials/Fortran_90_and_95/`

13.4.5 Open Directory - Fortran

`http://dmoz.org/Computers/Programming/Languages/Fortran/`

Chapter 14

Useful online resources and publications

Version 1.3, April 2020. Updated the Fortran Forum entry. Damian Rouson has taken over as editor.

Version 1.2, April 2020. Added an entry for the ACM publication Fortran Forum.

Version 1.1, June 2019. Renamed the chapter and added links to the IEEE, ACM and British Library

Version 1.0, January 2019. The chapter on other links has been split into 2.

14.1 Association for Computing Machinery - ACM

Here is an extract from their site.

> ACM brings together computing educators, researchers, and professionals to inspire dialogue, share resources, and address the field's challenges. As the world's largest computing society, ACM strengthens the profession's collective voice through strong leadership, promotion of the highest standards, and recognition of technical excellence. ACM supports the professional growth of its members by providing opportunities for life-long learning, career development, and professional networking.

Visit
`https://www.acm.org/`
for more information.
Visit
`https://dl.acm.org/`
for access to their digital library.
Here is a quote from that site.

> The ACM Digital Library is a research, discovery and networking platform containing:

> The Full-Text Collection of all ACM publications, including journals, conference proceedings, technical magazines, newsletters and books.
>
> A collection of curated and hosted full-text publications from select publishers.
>
> The ACM Guide to Computing Literature, a comprehensive bibliographic database focused exclusively on the field of computing.
>
> A richly interlinked set of connections among authors, works, institutions, and specialized communities.

Several of the publications in this resource file are available on their site.

14.1.1 Fortran Forum

Here is the home page.
`https://dl.acm.org/newsletter/sigplan-fortran`
Here is the text from the home page.

> SIGPLAN FORTRAN Forum (not included in membership) Addresses the FORTRAN language, its uses, profitability, standardization, further evolution, and the implementation of FORTRAN processors. Published 3 times per year.

Damian Rouson is the current editor.

14.2 British Library

The British Library
`https://www.bl.uk/`
is a very useful online resource. Here is an extract from their site.

> We are the national library of the United Kingdom and give access to the world's most comprehensive research collection. We provide information services to academic, business, research and scientific communities.
>
> Our collection of over 170 million items includes artifacts from every age of written civilisation. We keep the nation's archive of printed and digital publications, adding around three million new items to our collection every year.
>
> We have many books, but we have so much more. Our London and Yorkshire sites have everything from newspapers to sound recordings, patents, prints and drawings, maps and manuscripts. Our inspiring exhibitions interpret these collections and bring their stories to the public.

They have an online catalogue
`http://explore.bl.uk/primo_library/libweb/action/search.do?vid=BLVU1`
that is easily searchable and was used in providing details of ISBN's for books in this resource file.

14.3 Institute of Electrical and Electronics Engineers - IEEE

Here is an extract from their site

> IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

Visit
`https://www.ieee.org/`
for more information.
Visit
`https://ieeexplore.ieee.org/Xplore/home.jsp`
for access to their digital library. Several of the publications in this resource file can be found on their site.

14.4 Object Oriented Programming

Here is a link to a site that has a very good coverage of object oriented programming. Here is a snapshot of the content on that site.

> General Resources on Object Oriented Programming
>
> Online
>
> Books
>
> Object Modeling and UML
>
> OO Resources by Language
>
> - C-Based Languages
> - Common Lisp
> - Erlang
> - F#
> - Fortran
> - Go
> - Java
> - JavaScript
> - PHP
> - Python
> - Ruby
> - Smalltalk

On the Other Hand...

Bottom Line on OOP

http://www.whoishostingthis.com/resources/oo-programming/

14.5 Programming

14.5.1 Calling FORTRAN and C from Java

http://www.csharp.com/javacfort.html

14.5.2 CS 267 Applications of Parallel Computers

http://www.cs.berkeley.edu/~yozo/cs267.sp05/

14.5.3 Hillside.net - Design Patterns Book - DP Book

http://hillside.net/patterns/DPBook/DPBook.html

14.5.4 Hillside.net - Design Patterns Book - Source

http://hillside.net/patterns/DPBook/Source.html

14.5.5 Home page of Les Hatton

http://www.leshatton.org/

14.5.6 Parallel Programming - Basic Theory For The Unwary

http://users.actcom.co.il/~choo/lupg/tutorials/parallel-programming-theory/parallel-programming-theory.html

14.5.7 Putting a Java Interface on your C, C++, or Fortran Code

http://www.math.ucla.edu/~anderson/JAVAclass/JavaInterface/JavaInterface.html

14.5.8 Teach Yourself Programming in Ten Years

http://www.norvig.com/21-days.html

Chapter 15

Computer Arithmetic and IEEE 754 formats

Version 1.0, September 2020

15.1 Computer Arithmetic sources

Here are some sources.

15.1.1 What every computer scientist should know about floating point arithmetic

`http://www.validlab.com/goldberg/paper.pdf`

15.1.2 IEEE 754r - Wikipedia, the free encyclopedia

`http://en.wikipedia.org/wiki/IEEE_754r`

15.1.3 IEEE 754 Standard for Binary Floating-Point Arithmetic

`http://grouper.ieee.org/groups/754/`

15.1.4 IEEE Standard 754 Floating-Point

`http://stevehollasch.com/cgindex/coding/ieeefloat.html`

15.1.5 William Kahan

`http://www.cs.berkeley.edu/~wkahan/`

15.1.6 IEEE 754 floating-point test software

`http://www.math.utah.edu/~beebe/software/ieee/`

15.1.7 Interval FAQ from Alejandro Casares – What machines support IEEE 754

`http://www.mscs.mu.edu/~georgec/IFAQ/casares1.html`

15.1.8 Decimal Arithmetic - FAQ 1

`http://www2.hursley.ibm.com/decimal/decifaq1.html#emphasis`

15.1.9 General Decimal Arithmetic

`http://www2.hursley.ibm.com/decimal/`

15.2 IEEE arithmetic and IEEE formats

Wikipedia has a very good entry on IEEE arithmetic. The following has been taken from their site.

The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point arithmetic established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). The standard addressed many problems found in the diverse floating-point implementations that made them difficult to use reliably and portably. Many hardware floating-point units use the IEEE 754 standard.

The standard defines:

arithmetic formats: sets of binary and decimal floating-point data, which consist of finite numbers (including signed zeros and subnormal numbers), infinities, and special "not a number" values (NaNs)

interchange formats: encodings (bit strings) that may be used to exchange floating-point data in an efficient and compact form

rounding rules: properties to be satisfied when rounding numbers during arithmetic and conversions

operations: arithmetic and other operations (such as trigonometric functions) on arithmetic formats

exception handling: indications of exceptional conditions (such as division by zero, overflow, etc.)

IEEE 754-2008, published in August 2008, includes nearly all of the original IEEE 754-1985 standard, plus the IEEE 854-1987 Standard for Radix-Independent Floating-Point Arithmetic. The current version, IEEE 754-2019, was published in July 2019. It is a minor revision of the previous version, incorporating mainly clarifications, defect fixes and new recommended operations.

Visit
`https://en.wikipedia.org/wiki/IEEE_754`
for a detailed coverage.

Here is a table of the current IEEE arithmetic formats.

	Binary				
IEEE name	binary16	binary32	binary64	binary128	binary256
Common name	Half precision	Single precision	Double precision	Quadruple precision	Octuple precision
Base	2	2	2	2	2
Significand (bits of digits)	11	24	53	113	237
Decimal digits	3.31	7.22	15.95	34.02	71.34
Exponent bits	5	8	11	15	19
Decimal E max	4.51	38.23	307.95	4931.77	78913.2
Exponent bias	2**4-1	2**7-1	2**10-1	2**14-1	2**18-1
Exponent bias	15	127	1023	16383	262143
E min	-14	-126	-1022	-16382	-262142
E max	15	127	1023	16383	262143
Notes	not basic				not basic
	Decimal				
IEEE name	decimal32	decimal64	decimal128		
Common name					
Base	10	10	10		
Significand (bits of digits)	7	16	34		
Decimal digits	7	16	34		
Exponent bits	7.58	9.58	13.58		
Decimal E max	96	384	6144		
Exponent bias	101	398	6176		
Exponent bias	101	398	6176		
E min	-95	-383	-6143		
E max	96	384	6144		
Notes	not basic				

IEEE arithmetic formats

The following Fortran module is the one we use in helping to control precision in our Fortran programs.

```
module precision_module
  implicit none
!
! Updated with the release of Nag 7 which
! supports 16 bit reals.
!
! single, double, quad naming used by lapack.
! hence sp, dp, qp
!
! we have used hp as half precision
!
  integer, parameter :: hp = selected_real_kind( 3,   4)
  integer, parameter :: sp = selected_real_kind( 6,  37)
  integer, parameter :: dp = selected_real_kind(15, 307)
  integer, parameter :: qp = selected_real_kind(30, 291)
end module
```