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

- September 2015. Added an entry to our tables on compiler conformance to the Fortran 2003, 2008 and TS 29113 standards.

- August 2015. Added an entry in the compiler section on compiler comparison tables for Windows and Linux based Fortran compilers. These tables are maintained by Polyhedron Solutions.

- August 2015. Added Jorgensen Fortran 95 book (free pdf) from a thread on comp.lang.fortran

- 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.

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

1 Books

1.1 Fortran 2003 and 2008 - English ........................................ 14
1.2 Fortran 95 - English ....................................................... 15
1.3 Fortran 90 - English ....................................................... 16
1.4 English books on related topics ....................................... 18
1.5 Chinese ................................................................. 18
1.6 Dutch ................................................................. 18
1.7 Finnish ............................................................... 18
1.8 French ................................................................. 19
1.9 German ............................................................... 19
1.10 Italian ................................................................. 20
1.11 Japanese ............................................................. 20
1.12 Russian ............................................................... 20
1.13 Swedish .............................................................. 20

2 Compilers

2.1 Introduction ............................................................... 22
  2.1.1 Free compilers for general use .................................. 22
  2.1.2 Free for personal use ............................................... 22
  2.1.3 Free for educational use ......................................... 23
  2.1.4 Windows and Linux Compiler Comparison tables .......... 23
  2.1.5 Standards conformance ........................................... 24
2.2 Absoft ................................................................. 24
2.3 Cray ................................................................. 24
2.4 Fortran Company ......................................................... 24
2.5 Fujitsu ............................................................... 25
2.6 Gnu Fortran ............................................................ 25
2.7 g95 ................................................................. 26
2.8 Hewlett Packard ........................................................ 26
2.9 IBM ................................................................. 26
2.10 Intel ................................................................. 27
2.11 Lahey/Fujitsu .......................................................... 28
2.12 Lahey/GNU Shasta Compiler ....................................... 28
2.13 NAG ................................................................. 28
2.14 NEC ................................................................. 29
## CONTENTS

2.15 Nocturnal Aviation Software ............................................. 29
2.16 OpenUH ................................................................. 30
2.17 Oracle - originally Sun .................................................. 30
2.18 PathScale ................................................................. 31
2.19 PGI ................................................................. 31
2.20 Silverfrost, nee Salford Software ....................................... 31
2.21 SGI ................................................................. 32
  2.21.1 MIPSpro Fortran Compilers ........................................ 32
  2.21.2 MIPSpro Fortran 90 Compiler ..................................... 32
2.22 Sun - see Oracle ......................................................... 32
2.23 No longer available ......................................................... 32
  2.23.1 Apogee .......................................................... 32
  2.23.2 Compaq ........................................................ 33
  2.23.3 EPC ............................................................ 33
  2.23.4 NA Software ...................................................... 33

3 Debuggers 35

3.1 Introduction .............................................................. 35
3.2 Absoft ................................................................. 35
3.3 DDT ................................................................. 35
3.4 gnu gdb ............................................................ 36
3.5 Intel IDB ............................................................. 37
3.6 LLDB ................................................................. 37
3.7 Microsoft Visual Studio Debugger .................................... 37
3.8 totalview .............................................................. 37
3.9 Valgrind ............................................................... 38
3.10 WinDbg .............................................................. 39

4 Fortran aware editors and IDEs 41

4.1 Windows ................................................................. 41
  4.1.1 Absoft Editor (ae) .................................................. 41
  4.1.2 Code::Blocks ..................................................... 41
  4.1.3 CRiSP .......................................................... 41
  4.1.4 Compaq Visual Fortran 6.x ...................................... 41
  4.1.5 editeur .......................................................... 41
  4.1.6 emacs/xemacs - stand alone ..................................... 42
  4.1.7 emacs/xemacs - cygwin components ............................ 42
  4.1.8 geany ............................................................ 42
  4.1.9 gvim/vim - stand alone ......................................... 42
  4.1.10 gvim/vim - cygwin component ................................ 42
  4.1.11 jed, wjed (Windows) ........................................... 42
  4.1.12 lahey ed ........................................................ 42
  4.1.13 Microsoft Visual Studio 6 ...................................... 42
  4.1.14 Microsoft Visual Studio 2013 Community Edition ........... 42
  4.1.15 Microsoft Visual Studio.NET ................................. 43
## 4.1.16 nedit - cygwin

<table>
<thead>
<tr>
<th>4.1.17 ntemacs</th>
<th>43</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.18 photran</td>
<td>43</td>
</tr>
<tr>
<td>4.1.19 Salford plato</td>
<td>43</td>
</tr>
<tr>
<td>4.1.20 UltraEdit</td>
<td>43</td>
</tr>
<tr>
<td>4.1.21 xemacs/emacs - stand alone</td>
<td>44</td>
</tr>
<tr>
<td>4.1.22 xemacs/emacs - cygwin components</td>
<td>44</td>
</tr>
<tr>
<td>4.1.23 Zeus ide</td>
<td>44</td>
</tr>
</tbody>
</table>

## 4.2 Linux/Unix

<table>
<thead>
<tr>
<th>4.2.1 Code::Blocks</th>
<th>44</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2 CRiSP</td>
<td>44</td>
</tr>
<tr>
<td>4.2.3 emacs/xemacs</td>
<td>44</td>
</tr>
<tr>
<td>4.2.4 geany</td>
<td>44</td>
</tr>
<tr>
<td>4.2.5 ied, xied (Unix(all flavours)/OpenVMS) wied (Windows)</td>
<td>44</td>
</tr>
<tr>
<td>4.2.6 nedit</td>
<td>44</td>
</tr>
<tr>
<td>4.2.7 Oracle Solaris Studio Express</td>
<td>44</td>
</tr>
<tr>
<td>4.2.8 photran</td>
<td>44</td>
</tr>
</tbody>
</table>

## 4.3 Apple OS X

<table>
<thead>
<tr>
<th>4.3.1 Absoft Editor</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.2 BBEdit</td>
<td>45</td>
</tr>
<tr>
<td>4.3.3 codeblocks</td>
<td>45</td>
</tr>
<tr>
<td>4.3.4 emacs/xemacs</td>
<td>45</td>
</tr>
<tr>
<td>4.3.5 Photran</td>
<td>45</td>
</tr>
<tr>
<td>4.3.6 Smultron</td>
<td>45</td>
</tr>
<tr>
<td>4.3.7 TextMate</td>
<td>45</td>
</tr>
<tr>
<td>4.3.8 TextWrangler</td>
<td>45</td>
</tr>
<tr>
<td>4.3.9 Vim</td>
<td>46</td>
</tr>
<tr>
<td>4.3.10 Xcode</td>
<td>46</td>
</tr>
<tr>
<td>4.3.11 xemacs/emacs</td>
<td>46</td>
</tr>
</tbody>
</table>

## 5 Commercial Courses

<table>
<thead>
<tr>
<th>5.1 Archer</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2 Ian Chivers and Jane Sleightholme</td>
<td>48</td>
</tr>
<tr>
<td>5.3 Cranfield University</td>
<td>48</td>
</tr>
<tr>
<td>5.4 The Fortran Company</td>
<td>48</td>
</tr>
<tr>
<td>5.5 Laches</td>
<td>49</td>
</tr>
<tr>
<td>5.6 Michael Metcalf</td>
<td>49</td>
</tr>
<tr>
<td>5.7 Nihon NAG, Numerical Algorithms Group Japan</td>
<td>49</td>
</tr>
<tr>
<td>5.8 PTR Associates</td>
<td>49</td>
</tr>
<tr>
<td>5.9 Purple Sage Computing Solutions, Inc</td>
<td>49</td>
</tr>
<tr>
<td>5.10 Sourcery, Inc</td>
<td>50</td>
</tr>
<tr>
<td>5.11 France</td>
<td>50</td>
</tr>
<tr>
<td>5.12 Japan</td>
<td>50</td>
</tr>
<tr>
<td>5.12.1 Nihon NAG, Numerical Algorithms Group Japan</td>
<td>50</td>
</tr>
</tbody>
</table>
CONTENTS

6 On Line Training Material
6.1 CERN ................................................................. 51
6.2 Paul Dubois .......................................................... 51
6.3 Linkoping University ............................................... 51
6.4 Liverpool University ................................................ 51
6.5 French ................................................................. 52

7 Graphics and Windows Programming ......................... 53
7.1 Introduction .......................................................... 53
7.2 dislin ................................................................. 53
  7.2.1 Worked examples ................................................ 54
7.3 gino ................................................................. 54
  7.3.1 Documentation .................................................. 54
  7.3.2 Worked examples .............................................. 54
7.4 ginomenu ............................................................ 54
  7.4.1 Documentation .................................................. 54
7.5 interacter ............................................................ 55
  7.5.1 Documentation .................................................. 55
  7.5.2 Worked examples .............................................. 55
7.6 opengl ............................................................... 55
  7.6.1 Documentation .................................................. 55
  7.6.2 Worked examples .............................................. 55
  7.6.3 Documentation .................................................. 56
  7.6.4 Worked examples .............................................. 56
7.7 pgplot ............................................................... 56
7.8 realwin .............................................................. 57
  7.8.1 Documentation .................................................. 57
  7.8.2 Worked examples .............................................. 57
7.9 toolmaster .......................................................... 57
  7.9.1 Documentation .................................................. 57
  7.9.2 Worked examples .............................................. 57
7.10 winteracter ........................................................ 58
  7.10.1 Documentation .................................................. 58
  7.10.2 Worked examples .............................................. 58
7.11 Microsoft Windows graphics programming ................. 58
  7.11.1 Lahey/Fujitsu ............................................... 58
  7.11.2 Salford Software ............................................. 59

8 Parallel Programming .............................................. 61
8.1 Introduction ........................................................ 61
  8.1.1 Books ............................................................ 62
8.2 Automatic ........................................................... 63
8.3 Coarray Fortran .................................................... 63
  8.3.1 opencoarrays.org .............................................. 63
8.4 HPF ................................................................ 63
CONTENTS

8.5 MPI .......................................................... 63
  8.5.1 Books ...................................................... 64
  8.5.2 Courses ................................................... 64
  8.5.3 Requirements ............................................ 64

8.6 OpenMP ........................................................ 65
  8.6.1 Books ...................................................... 65
  8.6.2 Courses ................................................... 65
  8.6.3 Resources ................................................ 66
  8.6.4 Requirements .......................................... 66

8.7 Posix Threads .................................................. 66

8.8 Notes on the table below ...................................... 66

8.9 Table of compilers and supported parallel options .......... 68

8.10 Parallelisation Tools ........................................ 69
  8.10.1 Crescent Bay Software .................................. 69
  8.10.2 Parallel Software Products ............................... 70

8.11 Compiler and MPI combinations ................................ 70
  8.11.1 IBM ....................................................... 70
  8.11.2 Apple ...................................................... 70
  8.11.3 Nag ........................................................ 71
  8.11.4 PGI ......................................................... 71
  8.11.5 Oracle ..................................................... 71
  8.11.6 Intel ....................................................... 71

9 Fortran Software Tools ............................................. 73
  9.1 Refactoring .................................................. 73
  9.2 Convert ........................................................ 74
  9.3 Doctran ........................................................ 74
  9.4 Forcheck ....................................................... 74
  9.5 FOR_STRUCT .................................................. 75
  9.6 FOR_STUDY .................................................... 75
  9.7 Fortran90-lint ............................................... 75
  9.8 NAG ........................................................... 75
  9.9 photran ....................................................... 75
  9.10 plusFORT .................................................... 76
  9.11 VAST/77to90 ................................................ 76
  9.12 Visustin ..................................................... 76

10 Fortran Electronic Lists ......................................... 77
  10.1 comp-fortran-90 ............................................. 77
  10.2 comp.lang.fortran .......................................... 77
  10.3 LinkedIn ...................................................... 78
  10.4 Compiler specific ........................................... 78
    10.4.1 Absoft .................................................. 78
    10.4.2 Apogee .................................................. 78
    10.4.3 Compad ................................................ 79
10.4.4 Cray ................................................................. 79
10.4.5 Fortran Company .................................................. 79
10.4.6 Fujitsu ............................................................... 79
10.4.7 Gnu Fortran 95 ...................................................... 79
10.4.8 G95 ................................................................. 79
10.4.9 Hewlett Packard .................................................. 79
10.4.10 IBM ................................................................. 79
10.4.11 Intel ................................................................. 79
10.4.12 Lahey Fujitsu ..................................................... 80
10.4.13 NAG ................................................................. 80
10.4.14 NA Software ...................................................... 80
10.4.15 NEC ................................................................. 80
10.4.16 Pathscale ......................................................... 80
10.4.17 PGI ................................................................. 80
10.4.18 Salford Software ............................................... 80
10.4.19 SGI ................................................................. 80
10.4.20 SUN ............................................................... 80

11 Fortran Standard Bodies .................................................. 81
11.1 Introduction .......................................................... 81
11.2 WG5 .................................................................. 81
11.3 PL22.3 .................................................................. 82

12 Other Web Links ............................................................ 85
12.1 Fortran History ........................................................ 85
12.1.1 A brief history of FORTRAN-Fortran ....................... 85
12.1.2 Computer Languages History (preview) ................. 85
12.1.3 Computer Languages History ............................... 85
12.1.4 Fortran A few historical details ............................. 85
12.1.5 Open Directory - Fortran Tutorials Fortran 90 and 95 85
12.1.6 Open Directory - Fortran ..................................... 85
12.1.7 The Fortran (not the foresight) saga ...................... 86
12.2 Computer Arithmetic ................................................. 86
12.2.1 What every computer scientist should know about floating point arithmetic 86
12.2.2 IEEE 754r - Wikipedia, the free encyclopedia ........ 86
12.2.3 IEEE 754 Standard for Binary Floating-Point Arithmetic 86
12.2.4 IEEE Standard 754 Floating-Point ......................... 86
12.2.5 William Kahan .................................................... 86
12.2.6 IEEE 754 floating-point test software .................... 86
12.2.7 Interval FAQ from Alejandro Casares – What machines support IEEE 754 86
12.2.8 Decimal Arithmetic - FAO ................................. 86
12.2.9 General Decimal Arithmetic ............................... 86
12.3 Programming .......................................................... 87
12.3.1 Calling FORTRAN and C from Java ...................... 87
12.3.2 CS 267 Applications of Parallel Computers ............. 87
12.3.3  Hillside.net - Design Patterns Book - DP Book  . . . . . . . . . 87
12.3.4  Hillside.net - Design Patterns Book - Source  . . . . . . . . . 87
12.3.5  Home page of Les Hatton  . . . . . . . . . . . . . . . . . . . . 87
12.3.6  Parallel Programming - Basic Theory For The Unwary  . . . 87
12.3.7  Putting a Java Interface on your C, C++, or Fortran Code . . 87
12.3.8  Teach Yourself Programming in Ten Years  . . . . . . . . . . . 87
Chapter 1

Books

- Version 1.14, August 2015; Added Jorgenson Fortran 95 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 McCormack; 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;


1.1 Fortran 2003 and 2008 - English


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

and an additional set of index entries at


1.2. FORTRAN 95 - ENGLISH


1.2 Fortran 95 - English


- 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/


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


• Morgan & Schonfelder, Programming in Fortran 90/95, available in Acrobat PDF format from

http://www.fortran.com/


1.3 Fortran 90 - English


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


1.4 English books on related topics

  A CDROM is also available (see Web site).

1.5 Chinese


1.6 Dutch


1.7 Finnish

  http://www.csc.fi/oppaat/f95/
1.8 French


1.9 German


1.10 Italian

• Titolo: Programmare con Fortran versioni 90952003 Autore: Ciaburro Giuseppe

http://www.ibs.it/code/9788882336882/ciaburro-giuseppe/programmare-con-fortran.html

1.11 Japanese


1.12 Russian


1.13 Swedish

Chapter 2
Compilers


- Version 1.16, August 2015. Add a link to the Polyhedron Solutions web site. They have compiler comparison tables for Windows and Linux compilers.


- 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 1.12, 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.
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.

- **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.

- **Oracle -** Current Release - Oracle Solaris Studio 12.4 Provides perpetual no-cost license for production use and the development of commercial applications. You have to accept the license agreement.

http://www.oracle.com/technetwork/licenses/solaris-studio-license-169628.html

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. This 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.

2.1.4 Windows and Linux Compiler Comparison tables

Polyhedron Solutions is a software house based near Oxford in the UK. Their website provides compiler comparison tables for Windows and Linux Fortran compilers.

Their home address is

http://www.fortran.uk/

and the compiler comparison tables can be found at

http://www.fortran.uk/compiler-comparisons/

Articles on these tables also appear in the ACM Fortran Forum newsletter.

http://dl.acm.org/citation.cfm?id=J286&picked=prox&cfrid=537168128&cftoken=94354736

Here is an extract from their site describing these tables.

Which is the best Fortran compiler? Were often asked that question, but there is no single answer. All compilers have strong and weak points, and your choice should depend on finding the best match between your requirements and a compilers strengths. To help our customers make an informed decision, we have compiled a series of charts comparing the main contenders.

The Version Numbers of the compilers used in these comparisons are listed in the Benchmark Tables.

Windows Compilers

- Features  What other tools are supplied with the compiler?
- 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 Sandy Bridge Intel Core i5 2500k)  17 Fortran benchmarks

- Fortran Execution Time Benchmarks (64-bit Windows 7 on AMD Phenom II)  17 Fortran benchmarks

Linux Compilers

- Features  What other tools are supplied with the compiler?
- Language  What extensions to standard Fortran do the compilers support?
• Diagnostic Capabilities Can they spot programming errors?

• Fortran Execution Time Benchmarks (Linux64 on Sandy Bridge Intel Core i5 2500k) 17 Fortran benchmarks

• Fortran Execution Time Benchmarks (Linux64 on AMD Phenom II) 17 Fortran benchmarks

2.1.5 Standards conformance

There is a lag between the publication of a Fortran standard and compiler implementation of the features of that standard. The following link


has details of Fortran compiler standard conformance to the Fortran 2003, 2008 and TS 29113 standards.

2.2 Absoft

http://www.absoft.com/

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.

2.3 Cray

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 interoperating with C described in TS 29113.

2.4 Fortran Company

http://www.fortran.com/
2.5. FUJITSU

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.

2.5 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.

2.6 Gnu Fortran

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

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.
2.7 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.

2.8 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.

2.9 IBM


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 enable-
ment, 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 program-
mers 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.

2.10 Intel


Intel Fortran Compilers
Performance without compromise on Windows*, Linux* and OS X* Broad support
for current and previous Fortran standards, plus popular extensions Intel Math Kernel
Library included in suites Optional Rogue Wave* IMSL* Fortran Numerical Library
on Windows
Intel Parallel Studio XE 2015
Deliver top C++ and Fortran application performance with less effort Faster code:
Boost applications performance that scales on todays and next-gen processors Create
code faster: Utilize a toolset that simplifies creating fast, reliable parallel code
Available for Windows, Linux and OS/X.

Students, educators, academic researchers, and open source contributors may
qualify for Free Software Tools.

Academic Researcher - For unfunded research (research not funded by grants). Academic researchers doing unfunded research (research not funded by grants) and are in the United States with an *.edu email address.

Student - For current students at degree-granting institutions. Matriculated stu-
dents of a higher education institution defined as a public or private vocational school,
correspondence school, junior college, college, or university. Students cannot be paid
and/or compensated for software development. The products below are currently free
for your use under the terms of the Non-Commercial License.

Educator - For use in teaching curriculum. Professors and educators teaching par-
allelism and high performance computing techniques to students at degree-granting
institutions.
Open Source Contributor - For developers actively contributing to open source projects. Developers must be actively contributing to an open source project (e.g. GitHub). Agree to terms and conditions of the Intel Non-Commercial License.

2.11 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.

2.12 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!

2.13 NAG

http://www.nag.co.uk/nagware.asp
The NAG Fortran Compiler, derived from the world’s first Fortran 90 Compiler from NAG is robust, highly tested, and valued by developers all over the globe for its checking capabilities and detailed error reporting. Available on a wide range of Unix and Windows platforms it accepts fixed or free format Fortran 95 input and many common Fortran 77 extensions are allowed. A large number of Fortran 2003 language features are now available. HPF code is also compiled and checked though only single processor output is generated. The 5.3 release supports a good working subset of OpenMP 3.

The following versions are available.
Apple Intel Mac64 Release 6.0 OS X 10.6-10.9
Apple Intel Mac 32 Release 5.2 OS X 10.4-10.5
Apple PowerPC Mac Release 5.2 OS X 10.3-10.4
HP PA-RISC 1.1 Release 5.1 HP ANSI C
IBM Power AIX Release 5.1 AIX XL C 6.0
IBM Z9 Linux Release 5.1 gcc 3.4.6
IBM Z9 Open Edition Release 5.2 IBM C
Linux x86-32 Release 6.0
Linux64 Release 6.0
Linux(IA64), Iteanium, 5.2, glibc 2.3 or later.
Sun SPARC Solaris Release 5.2 gcc 3.4.2
Sun SPARC Solaris Release 5.3 Sun C v 5.9
x86 and x86-64 Windows Release 6.0
An integrated Fortran IDE is also available for Windows.

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

2.14 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.

2.15 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.

2.16 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.

2.17 Oracle - originally Sun


http://www.sun.com/

```
\begin{verbatim}
http://developers.sun.com/sunstudio/
http://developers.sun.com/sunstudio/downloads/express/
```

Sun Studio software delivers high-performance, optimizing C, C++, and Fortran compilers for the Solaris OS on SPARC, and both Solaris and Linux on x86/x64 platforms, including the latest multi-core systems.

What’s in Sun Studio:

All Compilers - Specify 32-bit or 64-bit Address Model, Auto-parallelization of single-threaded code, Flags, Static data-race and deadlock-detection for x86, Math and Visual Instruction Set Support in SPARC64 VI, Option for Floating-Point, Fused or Multiply-Add Instructions, Option for Thread Analyzer support, Linux Support, OpenMP and OpenMPI support, Support for the directives, clauses and attributes.
Fortran Compiler - Interval arithmetic support on Solaris Intel platform, Faster compilation time for source files, UNSIGNED arguments, Backward compatibility with f77

C Compiler - More options, Auto-parallelization of single-threaded code, GNU C compatibility

C++ Compiler - More options, ABI compatibility, Compiler flags to optimize for multicore architectures, Static linking the standard C++ runtime library.

2.18 PathScale

http://www.pathscale.com/products/
pathscale_compiler_suite

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.

2.19 PGI

http://www.pgroup.com/

High-performance Optimizing Parallel Compilers For multi-core 64-bit x64 and 32-bit x86 processor-based Linux, Mac OS X and Windows workstations, servers and clusters. PGI Workstation includes a single seat license for PGI’s suite of compilers and tools. Also available with a multi-user network floating license (PGI Server). The PGI CDK Cluster Development Kit includes all the software for building and programming a turn-key Linux cluster. PGI Visual Fortran fully integrates PGI parallel Fortran into Microsoft Windows using Microsoft Visual Studio 2005.

2.20 Silverfrost, nee Salford Software


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.
CHAPTER 2. COMPILERS

2.21 SGI


The following is taken from the SGI site.

2.21.1 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

2.21.2 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.

2.22 Sun - see Oracle

2.23 No longer available

2.23.1 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.

2.23.2 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.

2.23.3 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

2.23.4 NA Software

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

No longer available.
Chapter 3

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.

3.1 Introduction

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

3.2 Absoft

The following information was provided by Wood Lotz.

Absoft Fx3 debugger

It supports Fortran, C and asm 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

3.3 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.

3.4 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
3.5. **INTEL IDB**

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.)

3.6 **LLDB**

3.7 **Microsoft Visual Studio Debugger**

3.8 totalview

Visit


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:

3.9 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 Asche, 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.

3.10 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.
CHAPTER 3. DEBUGGERS

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 4

Fortran aware editors and IDEs

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

4.1 Windows

4.1.1 Absoft Editor (ae)
http://www.absoft.com

4.1.2 Code::Blocks
http://www.codeblocks.org/

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

4.1.4 Compaq Visual Fortran 6.x
No longer available

4.1.5 editeur
www.studiware.com
CHAPTER 4. FORTRAN AWARE EDITORS AND IDES

4.1.6 emacs/xemacs - stand alone
http://www.gnu.org/software/emacs/emacs.html

4.1.7 emacs/xemacs - cygwin components
http://www.cygwin.com/

4.1.8 geany
http://www.geany.org/

4.1.9 gvim/vim - stand alone
http://www.vim.org/

4.1.10 gvim/vim - cygwin component
http://www.cygwin.com/

4.1.11 jed. wjed (Windows)
http://www.jedsoft.org/jed/

4.1.12 lahey ed
http://www.lahey.com/

4.1.13 Microsoft Visual Studio 6
No longer available.

Microsoft have made Visual Studio 2013 Professional available as a 'Community Edition'.
Visit


for more details. It is a free download. It includes Microsoft C++, C#, Visual Basic and F#. Here are details of some of the licence 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.

### 4.1.15 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](http://www.intel.com)

- Lahey/Fujitsu Fortran
  - [www.lahey.com](http://www.lahey.com)

- Silverfrost Salford FTN95

### 4.1.16 nedit - cygwin

- [http://www.nedit.org/](http://www.nedit.org/)

### 4.1.17 ntemacs


### 4.1.18 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).

### 4.1.19 Salford plato


### 4.1.20 UltraEdit

- [www.ultraedit.com](http://www.ultraedit.com)
4.1.21 xemacs/emacs - stand alone
http://www.gnu.org/software/emacs/emacs.html

4.1.22 xemacs/emacs - cygwin components
http://www.cygwin.com/

4.1.23 Zeus ide
http://www.zeusedit.com/fortran.html

4.2 Linux/Unix

4.2.1 Code::Blocks
http://www.codeblocks.org/

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

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

4.2.4 geany
http://www.geany.org/

4.2.5 jed, xjed (Unix(all flavours)/OpenVMS) wjed (Windows)
http://www.jedsoft.org/jed/

4.2.6 nedit
http://www.nedit.org/

4.2.7 Oracle Solaris Studio Express
http://developers.sun.com/sunstudio/downloads/express/

   Requires a Java run time.
4.2.8 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.

4.3 Apple OS X

4.3.1 Absoft Editor

http://www.absoft.com/

4.3.2 BBEdit

macosx/productivity_tools/bbedit.html

4.3.3 codeblocks

http://www.codeblocks.org/

4.3.4 emacs/xemacs

pre-installed

4.3.5 Photran

www.photran.org/

4.3.6 Smultron

http://smultron.sourceforge.net/

4.3.7 TextMate

http://macromates.com/

4.3.8 TextWrangler

productivity_tools/textwrangler.html
4.3.9 Vim
Pre-installed

4.3.10 Xcode
http://developer.apple.com/

4.3.11 xemacs/emacs
pre-installed
Chapter 5

Commercial Courses


• 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.

5.1 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
CHAPTER 5. COMMERCIAL COURSES

Performance Programming
PGAS Programming
Programming the Xeon Phi

5.2 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

5.3 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/

- Intermediate Programming in Fortran 2003 (3 days)

  https://www.cranfield.ac.uk/courses/training/

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

5.4 The Fortran Company

http://www.fortran.com/}

Follow training links.
5.5 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.

5.6 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 at michael.metcalf@t-online.de, or Manfred von Richthofen Straße 15, 12101 Berlin, Germany, +0049.30.78952573.

5.7 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

5.8 PTR Associates

Currently offer two Fortran courses.

http://www.ptr.co.uk/fortran-conversion-course.html
http://www.ptr.co.uk/fortran-programming.html.

5.9 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


Also on offer is a one day workshop on the new features of Fortran 2000. See


for more details.
5.10 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)

5.11 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

5.12 Japan

5.12.1 Nihon NAG, Numerical Algorithms Group Japan
Ofers 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 6

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

6.1 CERN

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

6.2 Paul Dubois


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.)

6.3 Linkoping University

http://www.nsc.liu.se/~boein/f77to90/f77to90.html

Fortran 77 to 90 Converson Course

6.4 Liverpool University

http://www.liv.ac.uk/HPC/HPCpage.html

Covers f90 and HPF, with Java-enhanced Web pages.
6.5 French

Support de cours Fortran 90 IDRIS - Corde & Delouis

www.idris.fr/data/cours/lang/fortran/choix_doc.html
Chapter 7

Graphics and Windows Programming

- 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.

7.1 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.

7.2 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

53
CHAPTER 7. GRAPHICS AND WINDOWS PROGRAMMING

7.2.1 Worked examples

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

7.3 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, defence, 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

7.3.1 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.


7.3.2 Worked examples

None

7.4 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
7.5. INTERACTER

7.4.1 Documentation
Windows HTML Help, PDF and printed documentation

http://www.gino-graphics.com/support.html

7.5 interacter

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

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

7.5.1 Documentation
None

7.5.2 Worked examples
None

7.6 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/

7.6.1 Documentation
None

7.6.2 Worked examples
None

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 (search for OpenGL)

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.)

7.6.3 Documentation

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

7.6.4 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)

7.7 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
7.8. REALWIN

binding library handles conversion between C and Fortran argument-passing conventions.

PGPLOT has been tested with UNIX (most varieties, including Linux, SunOS, Solaris, HP/UX, 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.

7.8 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/

7.8.1 Documentation

None

7.8.2 Worked examples

None

7.9 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.

7.9.1 Documentation

None

7.9.2 Worked examples

http://www.avs.com/software/soft_t/toolm.html
7.10 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.

7.10.1 Documentation

None

7.10.2 Worked examples

None

7.11 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.


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.

7.11.1 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).
7.11.2 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 8

Parallel Programming

- 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 and 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.

8.1 Introduction

The Fortran language has been standardised a number of times

- Fortran 66
- Fortran 77
- Fortran 90
- Fortran 95
- Fortran 2003
- Fortran 2008
and Fortran 2008 was completed in 2010.

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.

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

- HPF
- MPI
- 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

8.1.1 Books


8.2 Automatic

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

8.3 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.

8.3.1 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 current developer version of GNU Fortran (GFortran), which is part of GCC 5, is OpenCoarrays-compatible.

8.4 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


Requires source code modification.

8.5 MPI

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

http://www-unix.mcs.anl.gov/mpi/
8.5.1 Books

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


8.5.2 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/

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

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

8.5.3 Requirements

Requires the installation of the MPI library (some compiler companies offer a bundle of compiler and MPI library) and source code modification.
8.6 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

8.6.1 Books


8.6.2 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/
8.6.3 Resources

http://www.openmp.org/wp/resources/

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

http://openmp.org/wp/openmp-specifications/

8.6.4 Requirements

Requires source code modification.

8.7 Posix Threads

Posix Threads is a library specification for multithreading, proposed as a standard by a broadly based committee of vendors, implementors, 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/

8.8 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

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.

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.

### 8.9 Table of compilers and supported parallel options

<table>
<thead>
<tr>
<th></th>
<th>Automatic</th>
<th>Co Array</th>
<th>HPF</th>
<th>MPI</th>
<th>OpenMP</th>
<th>Posix Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absoft</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>C</td>
</tr>
<tr>
<td>Cray</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVP</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>C</td>
</tr>
<tr>
<td>T3E</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>X1</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>C</td>
</tr>
<tr>
<td>Fujitsu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sparc</td>
<td>C</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>g95</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gfortran</td>
<td>P</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP-UX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tru64 Unix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Openvms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Openvms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Openvms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XL</td>
<td>Y</td>
<td></td>
<td></td>
<td>C</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Intel</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>Lahey/Fujitsu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 8.10 Parallelisation Tools

#### 8.10.1 Crescent Bay Software

Offer a range of parallelisation tools.

http://www.crescentbaysoftware.com/end_user.html

VAST-F/Parallel (for Fortran) and VAST-C/Parallel (for C), from Crescent Bay Software, are automatic parallelizing preprocessors that can significantly improve the performance of your important applications on shared memory parallel platforms.
VAST/toOpenMP from Crescent Bay Software is a parallelizing optimizer that adds OpenMP directives to Fortran programs. Translated programs have portable parallelism that can execute efficiently on a variety of SMP parallel systems from multi-cpu PCs to Supercomputers.

8.10.2 Parallel Software Products

Offer a tool to help parallelise Fortran 77, Fortran 90 or Fortran 95 code. Their web address is

The tool can generate either MPI calls or OpenMP calls.

8.11 Compiler and MPI combinations

The following is a list of combinations of compilers and MPI library versions that are known to work or have worked in the past. The list is not exhaustive, and other combinations do work.

8.11.1 IBM

<table>
<thead>
<tr>
<th>hardware:</th>
<th>IBM Power4 64-bit systems or compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td>operating system:</td>
<td>AIX v5.3 or compatible</td>
</tr>
<tr>
<td>Fortran compiler:</td>
<td>XL Fortran v9.1 or compatible</td>
</tr>
<tr>
<td>MPI:</td>
<td>IBM Parallel Operating Environment</td>
</tr>
<tr>
<td></td>
<td>v4.1.1 or compatible</td>
</tr>
</tbody>
</table>

8.11.2 Apple

<table>
<thead>
<tr>
<th>hardware:</th>
<th>Apple G4/G5 Power Macintosh</th>
</tr>
</thead>
<tbody>
<tr>
<td>operating system:</td>
<td>Mac OS X 10.3.9, Mac OS X 10.4.5</td>
</tr>
<tr>
<td>Fortran compiler:</td>
<td>GNU g77 3.4.5</td>
</tr>
<tr>
<td></td>
<td>(needs linker found in cctools-528.5.dmg or later)</td>
</tr>
<tr>
<td></td>
<td>see ftp://gcc.gnu.org/pub/gcc/infrastructure/)</td>
</tr>
<tr>
<td>C compiler:</td>
<td>Apple bundled gcc compiler (4.0.0 or 3.3)</td>
</tr>
<tr>
<td></td>
<td>or FSF gcc 3.4.5 or FSF gcc 4.0.2</td>
</tr>
<tr>
<td></td>
<td>or IBM XLC compiler 6.0</td>
</tr>
<tr>
<td></td>
<td>and compatible compilers</td>
</tr>
<tr>
<td>MPI:</td>
<td>MPICH2 version 1.0.3 compatible</td>
</tr>
</tbody>
</table>
8.11. COMPILER AND MPI COMBINATIONS

8.11.3 Nag

- **hardware:** AMD64
- **operating system:** 64-bit Linux
- **Fortran compiler:** NAGWare f95 Release 5.0 and compatible
- **C compiler:** gcc 3.3 and compatible
- **MPI:** MPICH 1.2.6 and compatible

8.11.4 PGI

- **hardware:** AMD64
- **operating system:** 64-bit Linux
- **Fortran compiler:** Portland Group pgf77 6.0-5 and compatible
- **C compiler:** Portland Group pgcc 6.0-5 and compatible
- **MPI:** MPICH 1.2.6 and compatible

8.11.5 Oracle

- **hardware:** Sun Fire V890+, UltraSPARC IV+, 1500MHz, 24 cores
- **operating system:** Solaris 10 Update 3
- **Fortran compiler:** Sun Studio 11 Fortran 95 8.2
- **compiler options:** -fsimple=1 -xvector=no -ftrap=%none -KPIC -xarch=v9a -dalign -stackvar -O4
- **C compiler:** Sun Studio 11 C Compiler 5.8
- **compiler options:** -DAdd_ -xarch=v9a -KPIC -O
- **MPI:** Sun ClusterTools 7
- **BLAS:** Sun Studio 11 Sun Performance Library

8.11.6 Intel

- **hardware:** Intel EM64T systems and compatible (including AMD64)
- **operating system:** Linux for x86-64 with glibc-2.5 and compatible
- **Fortran compiler:** Intel(R) Fortran Intel(R) 64 Compiler Version 11.0 and compatible
- **C compiler:** Intel(R) C Intel(R) 64 Compiler Version 11.0 and compatible
- **MPI:** Intel(R) MPI Library 3.2 for Linux
- **ScaLAPACK:** Intel(R) Math Kernel Library 10.1 for Linux
- **BLAS/LAPACK:** Intel(R) Math Kernel Library 10.1 for Linux
Chapter 9

Fortran Software Tools


- 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 1.2, 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.

9.1 Refactoring

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


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.


9.2 Convert

Fortran 77 to Fortran 90 converter by Mike Metcalf.

http://www.nag.co.uk/nagware/Examples/convert.f90

9.3 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/

9.4 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.
9.5. **FOR_STRUCT**

- is sold with full guarantee and support.

http://www.forcheck.nl/

9.5 **FOR_STRUCT**

http://www.cobalt-blue.com/fs/fsmain.htm

Restructures FORTRAN into Clean, Maintainable Code.

9.6 **FOR_STUDY**

Analyzes and Documents your FORTRAN code.

http://www.cobalt-blue.com/

9.7 **Fortran90-lint**

For Fortran 90 program analysis

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

9.8 **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/

9.9 **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/
9.10 plusFORT

The following is taken from their web 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.

http://www.polyhedron.com/

9.11 VAST/77to90

Fortran 77 to Fortran 90 translator

http://www.crescentbaysoftware.com/vast_77to90.html

9.12 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.


http://www.aivosto.com/visustin.html
Chapter 10

Fortran Electronic Lists


10.1 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

10.2 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 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

10.3 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.

10.4 Compiler specific

Some of the compiler suppliers provide electronic list support. Some provide an email address for technical support.

10.4.1 Absoft

http://forums.absoft.com/

support@absoft.com

10.4.2 Apogee

info@apogee.com
10.4.3 Compaq

10.4.4 Cray

10.4.5 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

10.4.6 Fujitsu

10.4.7 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

10.4.8 G95
http://groups.google.com/group/gg95

10.4.9 Hewlett Packard
vf-support@hp.com

10.4.10 IBM
Requires registration.

10.4.11 Intel
Requires registration.

10.4.12 Lahey Fujitsu
Requires registration.
http://www.lahey.com/support.htm
http://www.laheyforum.com/

10.4.13 NAG
Requires registration.
support@nag.co.uk

10.4.14 NA Software

10.4.15 NEC
http://www.nec.com/global/support/index.html

10.4.16 Pathscale
Requires registration.
http://pathscale.com/support.html
support@pathscale.com

10.4.17 PGI
Various offerings.
http://www.pgroup.com/support/index.htm

10.4.18 Salford Software
Various options.
http://www.silverfrost.com/22/ftn95/support/index.asp

10.4.19 SGI
Various options.
http://www.sgi.com/support/customerservice.html

10.4.20 SUN
http://forums.sun.com/category.jspa?categoryID=113
Chapter 11

Fortran Standard Bodies

- Version 1.1, August 2012; Updated J3 entry to reflect organisation within the US
- Version 1.0, January 2006.

11.1 Introduction

There are two main Fortran standards bodies and these are WG5 and PL22.3 (formerly J3). Each is covered in turn below.

11.2 WG5

Their home page is:-

http://www.nag.co.uk/sc22wg5/

Their document ftp site is at:-

ftp://ftp.nag.co.uk/sc22wg5/
\end{verbatim}

The following

\begin{verbatim}http://www.nag.co.uk/sc22wg5/links.html\end{verbatim}

has links to documents relating to the following standards: Fortran 66, Fortran 77, Fortran 90, Fortran 95, Fortran 2003 and Fortran 2008.
11.3 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

Current participants (August 2012) include

Bierman, Emeritus
Brainerd, Emeritus
Cray Inc, Bill Long
Hendrickson, Emeritus
Hirchert, Emeritus
IBM Corporation, Daniel Chen
Intel Corporation, Stan Whitlock
Jet Propulsion Laboratory, Van Snyder
Lahey, Emeritus
Maine, Emeritus
Martin, Emeritus
Meissner, Emeritus
National Center for Atmospheric Research, Emeritus
Oracle, Robert Corbett
Smith, Emeritus
The Numerical Algorithms Group Ltd, Malcolm Cohen
United States Dept of Energy, Emeritus
Wagener, Emeritus

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 (August 2012) is given below.

\begin{verbatim}
Voting Principals
=================
Daniel Chen, IBM Corp.
Malcolm Cohen, Nihon Numerical Algorithms Group K.K.
\end{verbatim}
Robert Corbett, Oracle America
William Long, Cray Inc.
Dan Nagle, Chair PL22.3
Craig E. Rasmussen, Los Alamos National Laboratory
Van Snyder, Jet Propulsion Laboratory
Stan Whitlock, Intel Corporation

Voting alternatives
===================

Aleksandar Donev - Craig Rasmussen
Steve Lionel - Stan Whitlock
Kelvin Li - Daniel Chen
Lorri Menard - Stan Whitlock
Toon Moene - Craig Rasmussen
John K. Reid - Dan Nagle
Damian Rouson - Craig Rasmussen
C. R. Schult - William Long
Dr. John Wallin - Dan Nagle
Rafik Zurob - Daniel Chen

Their version of the working draft is at:

Chapter 12

Other Web Links

- Version 1.2, June 2009; Updated several web links.
- Version 1.1, September 2008
- Version 1.0, January 2006

12.1 Fortran History

12.1.1 A brief history of FORTRAN-Fortran
http://www.ibiblio.org/pub/languages/fortran/ch1-1.html

12.1.2 Computer Languages History (preview)
http://www.levenez.com/lang/history.html

12.1.3 Computer Languages History
http://www.levenez.com/lang/

12.1.4 Fortran A few historical details
No longer available.

12.1.5 Open Directory - Fortran Tutorials Fortran 90 and 95
http://www.dmoz.org/Computers/Programming/Languages/Fortran/Tutorials/Fortran_90_and_95/

12.1.6 Open Directory - Fortran
http://dmoz.org/Computers/Programming/Languages/Fortran/
12.1.7 The Fortran (not the foresight) saga
http://www.fortranplus.co.uk/resources/brian_meeks_fortran_saga.pdf

12.2 Computer Arithmetic

12.2.1 What every computer scientist should know about floating point arithmetic

12.2.2 IEEE 754r - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/IEEE_754r

12.2.3 IEEE 754 Standard for Binary Floating-Point Arithmetic
http://grouper.ieee.org/groups/754/

12.2.4 IEEE Standard 754 Floating-Point
http://stevehollasch.com/cgindex/coding/ieeefloat.html

12.2.5 William Kahan
http://www.cs.berkeley.edu/~wkahan/

12.2.6 IEEE 754 floating-point test software
http://www.math.utah.edu/~beebe/software/ieee/

12.2.7 Interval FAQ from Alejandro Casares – What machines support IEEE 754
http://www.mscs.mu.edu/~georgec/IFAQ/casares1.html

12.2.8 Decimal Arithmetic - FAQ 1
http://www2.hursley.ibm.com/decimal/decifaq1.html#emphasis

12.2.9 General Decimal Arithmetic
http://www2.hursley.ibm.com/decimal/
12.3 Programming

12.3.1 Calling FORTRAN and C from Java
http://www.csharp.com/javacfort.html

12.3.2 CS 267 Applications of Parallel Computers
http://www.cs.berkeley.edu/~yozo/cs267.sp05/

12.3.3 Hillside.net - Design Patterns Book - DP Book
http://hillside.net/patterns/DPBook/DPBook.html

12.3.4 Hillside.net - Design Patterns Book - Source
http://hillside.net/patterns/DPBook/Source.html

12.3.5 Home page of Les Hatton
http://www.leshatton.org/

12.3.6 Parallel Programming - Basic Theory For The Unwary
http://users.actcom.co.il/~choo/lupg/tutorials/
parallel-programming-theory/parallel-programming-theory.html

12.3.7 Putting a Java Interface on your C, C++, or Fortran Code
http://www.math.ucla.edu/~anderson/JAVAclass/
JavaInterface/JavaInterface.html

12.3.8 Teach Yourself Programming in Ten Years
http://www.norvig.com/21-days.html