The Fortran Programming Language

This document provides a short overview of Fortran related resources, including a description of how the language is supported for use on LRZ systems via compilers, possibilities for consulting, and courses.

- Introductory remarks
- Language revisions and extensions
- Support and Services at LRZ
  - Availability of compilers
  - Consulting
  - Courses
- Fortran References
  - Books
  - Web resources
  - Newsgroups / Mailing Lists
  - Standardization

### Introductory remarks

Fortran (which is a contraction of “FORmula TRANslator”) is a programming language targeted at solving computational problems from science and engineering. It was developed in the 1950s inside IBM by a team led by John Backus; as a compiled language, it is designed for achieving the best possible speed for the generated executable code. Newer revisions of the Fortran standard are nowadays the basis for implementations of Fortran compilers by multiple vendors, which are typically deployed on computing systems ranging from workstations to large-scale HPC systems. Notwithstanding the advent of C, C++, Java and scripting languages like Perl or Python, Fortran is still quite extensively used for tasks which involve large-scale data processing or number crunching. Because modern compilers typically support creation of dynamic libraries, plug-ins written in Fortran can be embedded into applications written in other languages.

### Language revisions and extensions

The following table provides an overview of relevant language revisions. In some cases language extensions were separately published as Technical Report (TR) or Technical Specification (TS); these can be considered "mini-standards" whose content is usually integrated into the regular language standard once a new revision of it is prepared.

<table>
<thead>
<tr>
<th>Informal revision name</th>
<th>Year of publication</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran 77</td>
<td>1977</td>
<td>Many legacy libraries still use this version of the language; later revisions for the most part still support all the syntax and semantics of this version. However, newly developed programs should avoid using many old-fashioned features of this revision (especially fixed form and the global data concept).</td>
</tr>
<tr>
<td>Fortran 90</td>
<td>1990</td>
<td>Major revision of the language that brought many new features (especially array processing, an extended type system, free source form, block constructs, dynamic memory management, and modules).</td>
</tr>
<tr>
<td>Fortran 95</td>
<td>1995</td>
<td>Minor revision that added corrections and some features targeted at parallel processing. Some Fortran 77 features are declared obsolescent.</td>
</tr>
<tr>
<td>TR 15580</td>
<td>1999</td>
<td>Technical Report that added IEEE floating point support, including limited exception handling. This TR was integrated into the Fortran 2003 standard and is therefore obsolete.</td>
</tr>
<tr>
<td>TR 15581</td>
<td>1999</td>
<td>Technical Report that extended the semantics of ALLOCATABLE entities. This TR was also integrated into the Fortran 2003 standard.</td>
</tr>
<tr>
<td>Fortran 2003</td>
<td>2004</td>
<td>Major revision. The most relevant new features were support for object orientation, (limited) interoperability with C, and extensions for I/O processing.</td>
</tr>
<tr>
<td>TR 19767</td>
<td>2003</td>
<td>Technical Report that specified syntax and semantics of submodules. This TR was integrated into the Fortran 2008 standard.</td>
</tr>
<tr>
<td>Fortran 2008</td>
<td>2010</td>
<td>A moderate-sized revision that especially added a baseline set of parallel programming facilities (coarrays). This is what compilers currently support.</td>
</tr>
<tr>
<td>TS 29113</td>
<td>2012</td>
<td>Technical Specification that significantly extended the interoperaction with C semantics. The main purpose was to permit specification and implementation of a conforming MPI interface. This TS has been integrated into the Fortran 2018 standard and will be formally retired soon.</td>
</tr>
<tr>
<td>TS 18508</td>
<td>2015</td>
<td>Technical Specification that adds further parallel semantics to the coarray programming model. An overview of this was given at the workshop on Recent Advances in Parallel Programming - a PDF of the talk’s slides is available from the linked page. This TS has been integrated into the Fortran 2018 standard and will be formally retired soon.</td>
</tr>
<tr>
<td>Fortran 2018</td>
<td>2018</td>
<td>A minor revision that is mostly targeted at removing inconsistencies. The TS 29113 and TS 18508 have also been integrated. The currently valid standard.</td>
</tr>
</tbody>
</table>
Support and Services at LRZ

Availability of compilers

On the (Linux-based) HPC systems at LRZ, the following compilers are available:

- Intel compiler: This is the current mainline compiler
- GNU Fortran
- NAG compiler
- PGI compiler

Please consult the linked documentation page of each compiler for details on configuration and usage.

A campus license program for the Intel compiler exists at LRZ that also includes the Windows OS (Intel Visual Fortran) via a MS Visual Studio extension, and a MacOS version.

Consulting

If you have questions related to Fortran programming, please open a ticket with appropriate topic and problem description at our Service Desk.

Courses

Fortran workshops of 3 and 5 days duration, respectively, are held at LRZ twice a year; please look at the course schedule for details such as topics and dates. The course materials are also available.

Fortran References

Books

- Metcalf, Reid, and Cohen: Modern Fortran Explained, Oxford University Press, 2011 (describes Fortran 2008). Errata files are also available.

Web resources

- Fortran on Wikipedia.
- Fortran 95 Language features. A comprehensive overview by Michael Metcalf.
- Einarsson, Bo und Shokin, Yurij: Fortran 90 for the Fortran 77 Programmer. A web tutorial.
- Fortran pages of the Open Directory Project.
- SIGPLAN Fortran Forum (published 3 times per year).

Here some information about compilers not provided on the LRZ HPC systems:

- Open64 (open-sourced by SGI)
- g95 (Andy Vaught's gcc-derived compiler. Appears to have stalled development after 2012)
- IBM xlf (targeted at POWER-based systems)
- Cray (select an entry in the Supercomputing Category and search from there)
- Oracle Studio (used to be Sun Studio) provides Fortran compilers for Solaris and Linux

John Reid has written a number of papers that summarize language features in newer revisions:

- the new features of Fortran 2003
- the new features of Fortran 2008
- coarrays in Fortran 2008
- the new features of Fortran 2018

Newsgroups / Mailing Lists

- comp.lang.fortran on Google Groups
- Fortran 90 list on JISCmail
Standardization

The standardization effort is driven by two committees: The international working group ISO/IEC JTC1/SC22/WG5 that is responsible for establishing the work items, and evaluating them when they're done, and the US Fortran standards committee J3 (formally INCITS/PL22.3) that does the actual technical development.

- The WG5 web space and its documents.
- The J3 web space and its documents.

The current Fortran standard document can be purchased from ISO. The same applies for the currently active Technical Specifications.