1 INTRODUÇÃO

"I don't know what the technical characteristics of the standard language for scientific and engineering computation in the year 2000 will be... but I know it will be called Fortran."
John Backus

Esta apostila destina-se ao aprendizado da **Linguagem de Programação Fortran Moderno.** Por *Fortran Moderno (Modern Fortran)* entende-se a linguagem de programação de acordo com o seu padrão mais recente, conforme será brevemente apresentado nas seções a seguir.

1.1 AS ORIGENS DA LINGUAGEM FORTRAN

Programação no período inicial do uso de computadores para a solução de problemas em física, química, engenharia, matemática e outras áreas da ciência era um processo complexo e tedioso ao extremo. Programadores necessitavam de um conhecimento detalhado das instruções, registradores, endereços de memória e outros constituintes da *Unidade Central de Processamento* (CPU[1]) do computador para o qual eles escreviam o código. O *Código-Fonte* era escrito em um notação numérica denominada *código de máquina.* Com o tempo, códigos mnemônicos foram introduzidos, uma forma de programação conhecida como *código numérico* ou *código Assembler*. Estes códigos eram traduzidos em instruções para a CPU por programas conhecidos como *Assemblers.* Durante os anos 50 ficou claro que esta forma de programação era de todo inconveniente, no mínimo devido ao tempo necessário para se escrever e testar um programa, embora esta forma de programação possibilitasse um uso otimizado dos recursos da CPU.[2]

Estas dificuldades motivaram que um time de programadores da IBM, liderados por John Warner Backus (1924–2007), desenvolvessem uma das primeiras *linguagem de alto-nível*, denominada FORTRAN (FORmula TRANslation). Seu objetivo era produzir uma linguagem que fosse simples de ser entendida e usada, mas que gerasse um código numérico quase tão eficiente quanto a linguagem Assembler. Desde o início, o Fortran era tão simples de ser usado que era possível programar fórmulas matemáticas quase como se estas fossem escritas de forma simbólica. Isto permitiu que programas fossem escritos mais rapidamente que antes, com somente uma pequena perda de eficiência no processamento, uma vez que todo cuidado era dedicado na construção do *compilador*, isto é, no programa que se encarrega de traduzir o código-fonte em Fortran para código Assembler ou octal.[3]

Mas o Fortran foi um passo revolucionário também porque o programadores foram aliviados da tarefa tediosa de usar Assembler, assim concentrando-se mais na solução do problema em questão. Mais importante ainda, computadores se tornaram acessíveis a qualquer cientista ou engenheiro disposto a devotar um pequeno esforço na aquisição de um conhecimento básico em Fortran; a tarefa da programação não estava mais restrita a um *corpus* pequeno de programadores especialistas.

O Fortran disseminou-se rapidamente, principalmente nas áreas da física, engenharia e matemática, uma vez que satisfazia uma necessidade premente dos cientistas. Inevitavelmente,

[1] Do inglês: *Central Processing Unit.*

[2] Em `http://asm.sourceforge.net/intro/hello.html` pode-se visualizar o código `hello.asm`, o qual implementa em assembler, em um total de 13 linhas, o mesmo tipo de programa que pode ser escrito em três ou quatro linhas em `Fortran`. Veja a listagem 2.1.

[3] Diversas publicações comemorando a história do Fortran e de John Backus são disponíveis. Ver, por exemplo, John Backus - the Father of Fortran, History of FORTRAN and FORTRAN II, History of Fortran ou From FORTRAN II to Modern Fortran.

dialetos da linguagem foram desenvolvidos, os quais levaram a problemas quando havia necessidade de se trocar programas entre diferentes computadores. O dialeto de Fortran otimizado para processadores fabricados pela IBM, por exemplo, facilmente gerava erro quando se tentava rodar o mesmo programa em um processador Burroughs, ou em outro qualquer. Assim, em 1966, após quatro anos de trabalho, a Associação Americana de Padrões (*American Standards Association*), posteriormente Instituto Americano Nacional de Padrões (*American National Standards Institute*, ou ANSI) originou o primeiro padrão para uma linguagem de programação, agora conhecido como Fortran 66. Essencialmente, era uma subconjunto comum de vários dialetos, de tal forma que cada dialeto poderia ser reconhecido como uma extensão do padrão. Aqueles usuários que desejavam escrever programas *portáveis* deveriam evitar as extensões e restringir-se ao padrão.

O Fortran trouxe consigo vários outros avanços, além de sua facilidade de aprendizagem combinada com um enfoque em execução eficiente de código. Era, por exemplo, uma linguagem que permanecia próxima (e explorava) o hardware disponível, ao invés de ser um conjunto de conceitos abstratos. Ela também introduziu, através das declarações `COMMON` e `EQUIVALENCE`, a possibilidade dos programadores controlarem a alocação da armazenagem de dados de uma forma simples, um recurso que era necessário nos primórdios da computação, quando havia pouco espaço de memória, mesmo que estas declarações sejam agora consideradas potencialmente perigosas e tenham o seu uso desencorajado. Finalmente, o código-fonte permitia espaços em branco na sua sintaxe, liberando o programador da tarefa de escrever código em colunas rigidamente definidas e permitindo que o corpo do programa fosse escrito da forma desejada e visualmente mais atrativa.

A proliferação de dialetos permaneceu um problema mesmo após a publicação do padrão Fortran 66. A primeira dificuldade foi que muitos compiladores não aderiram ao padrão. A segunda foi a implementação, em diversos compiladores, de recursos que eram essenciais para programas de grande escala, mas que eram ignorados pelo padrão. Diferentes compiladores implementavam estes recursos de formas distintas.

Esta situação, combinada com a existência de falhas evidentes na linguagem, tais como a falta de construções estruturadas de programação, resultaram na introdução de um grande número de *pré-processadores.* Estes eram programas que eram capazes de ler o código fonte de algum dialeto bem definido de Fortran e gerar um segundo arquivo com o texto no padrão, o qual então era apresentado ao compilador nesta forma. Este recurso provia uma maneira de estender o Fortran, ao mesmo tempo retendo a sua portabilidade. O problema era que embora os programas gerados com o uso de um pré-processador fossem portáveis, podendo ser compilados em diferentes computadores, o código gerado era muitas vezes de uma dificuldade proibitiva para a leitura direta.

Estas dificuldades foram parcialmente removidas pela publicação de um novo padrão, em 1978, conhecido como Fortran 77. Ele incluía diversos novos recursos que eram baseados em extensões comerciais ou pré-processadores e era, portanto, não um subconjunto comum de dialetos existentes, mas sim um novo dialeto por si só. O período de transição entre o Fortran 66 e o Fortran 77 foi muito mais longo que deveria, devido aos atrasos na elaboração de novas versões dos compiladores e os dois padrões coexistiram durante um intervalo de tempo considerável, que se estendeu até meados da década de 80. Eventualmente, os fabricantes de compiladores passaram a liberá-los somente com o novo padrão, o que não impediu o uso de programas escritos em Fortran 66, uma vez que o Fortran 77 permitia este código antigo por compatibilidade. Contudo, diversas extensões não foram mais permitidas, uma vez que o padrão não as incluiu na sua sintaxe.

1.2 O padrão Fortran 90

Após trinta anos de existência, Fortran não mais era a única linguagem de programação disponível para os programadores. Ao longo do tempo, novas linguagens foram desenvolvidas e, onde elas se mostraram mais adequadas para um tipo particular de aplicação, foram adotadas em seu lugar. A superioridade do Fortran sempre esteve na área de aplicações numéricas, científicas, técnicas e de engenharia. A comunidade de usuários do Fortran realizou um investimento gigantesco em códigos, com muitos programas em uso frequente, alguns com centenas de milhares ou milhões de linhas de código. Isto não significava, contudo, que a comunidade estivesse completamente satisfeita com a linguagem. Vários programadores passaram a migrar seus có-

digos para linguagens tais como Pascal, C e C++. Para levar a cabo mais uma modernização da linguagem, o comitê técnico X3J3 (agora PL22.3), constituído pela ANSI, trabalhando como o corpo de desenvolvimento do comitê da ISO (*International Standards Organization*, Organização Internacional de Padrões) ISO/IEC JTC1/SC22/WG5 (doravante conhecido como WG5), preparou um novo padrão, inicialmente conhecido como Fortran 8x, e agora como Fortran 90. Os comitês X3J3 e WG5 são formados por representantes dos vendedores e fabricantes de hardware e desenvolvedores de software, academia e usuários da linguagem.

Quais eram as justificativas para continuar com o processo de revisão do padrão da linguagem Fortran? Além de padronizar extensões comerciais, havia a necessidade de modernização, em resposta aos desenvolvimentos nos conceitos de linguagens de programação que eram explorados em outras linguagens tais como APL, Algol, Pascal, Ada, C e C++. Com base nestas, o X3J3 podia usar os óbvios benefícios de conceitos tais como ocultamento de dados. Na mesma linha, havia a necessidade de fornecer uma alternativa à perigosa associação de armazenagem de dados, de abolir a rigidez agora desnecessária do formato fixo de fonte, bem como de aumentar a segurança na programação. Para proteger o investimento em Fortran 77, todo o padrão anterior foi mantido como um subconjunto do Fortran 90.

Contudo, de forma distinta dos padrões prévios, os quais resultaram quase inteiramente de um esforço de padronizar práticas existentes, o Fortran 90 é muito mais um desenvolvimento da linguagem, na qual são introduzidos recursos que são novos em Fortran, mas baseados em experiências em outras linguagens. Os recursos novos mais significativos são a habilidade de manipular matrizes usando uma notação concisa mais poderosa e a habilidade de definir e manipular tipos de dados definidos pelo programador. O primeiro destes recursos leva a uma simplificação na escrita de muitos problemas matemáticos e também torna o Fortran uma linguagem mais eficiente em supercomputadores. O segundo possibilita aos programadores a expressão de seus problemas em termos de tipos de dados que reproduzem exatamente os conceitos utilizados nas sua elaborações.

O padrão Fortran 90 foi publicado em 1991 e está disponível em https://wg5-fortran.org/N001-N1100/N692.pdf. Um relato pessoal das atividades do X3J3, inclusive citando resistências para a elaboração e adoção de um novo padrão, pode ser lido em https://www.nag.co.uk/content/personal-history-nag-fortran-compiler.

1.2.1 PROCEDIMENTOS ADOTADOS PARA A EVOLUÇÃO DA LINGUAGEM

Os procedimentos de trabalho adotados pelo comitê X3J3 estabelecem um período de aviso prévio antes que qualquer recurso existente seja removido da linguagem. Isto implica, na prática, um ciclo de revisão, que para o Fortran é de cerca de cinco anos. A necessidade de remoção de alguns recursos é evidente; se a única ação adotada pelo X3J3 fosse de adicionar novos recursos, a linguagem se tornaria grotescamente ampla, com muitos recursos redundantes e sobrepostos. Assim, além de detalhar os novos recursos que passam a ser adicionados à linguagem, cada padrão também possui um apêndice contendo duas listas:

Recursos obsolescentes (*obsolescent features*): tratam-se daqueles recursos ainda suportados pelo padrão, mas que são considerados obsoletos e, assim, candidatos à remoção no próximo ciclo de revisão.

Recursos removidos (*deleted features*): recursos que não são mais suportados na linguagem.

1.2.2 RECURSOS NOVOS DO FORTRAN 90

Um resumo dos novos recursos é dado a seguir:

- Operações de matrizes.
- Ponteiros.
- Recursos avançados para computação numérica usando um conjunto de funções inquisidoras numéricas.

- Parametrização dos tipos intrínsecos, permitindo o suporte a inteiros curtos, conjuntos de caracteres muito grandes, mais de duas precisões para variáveis reais e complexas e variáveis lógicas agrupadas.
- Tipos de dados derivados, definidos pelo programador, compostos por estruturas de dados arbitrárias e de operações sobre estas estruturas.
- Facilidades na definição de coletâneas denominadas *módulos*, úteis para definições globais de dados e para bibliotecas de subprogramas.
- Exigência que o compilador detecte o uso de construções que não se conformam com a linguagem ou que estejam em obsolescência.
- Um novo formato de fonte, adequado para usar em um terminal.
- Novas estruturas de controle, tais como `SELECT CASE` e uma nova forma para os laços `DO`.
- A habilidade de escrever subprogramas internos e subprogramas recursivos e de chamar subprogramas com argumentos opcionais.
- Alocação dinâmica de dados (matrizes automáticas, matrizes alocáveis e ponteiros).
- Melhoramentos nos recursos de entrada/saída de dados.
- Vários novos subprogramas intrínsecos.

Todos juntos, estes novos recursos contidos em Fortran 90 irão assegurar que o padrão continue a ser bem sucedido e usado por um longo tempo. O Fortran 77 continua sendo suportado como um subconjunto durante um período de adaptação.

1.2.3 Recursos em obsolescência no Fortran 90

Os recursos obsolescentes no Fortran 90 são:

- `IF` aritmético;
- desvio para uma declaração `END IF` a partir de um ponto fora de seu bloco;
- variáveis reais e de dupla precisão nas expressões de controle de um comando `DO`;
- finalização compartilhada de blocos `DO`, bem como finalização por uma declaração ou comando distintos de um `CONTINUE` ou de um `END DO`;
- declaração `ASSIGN` e comando `GO TO` atribuído;
- `RETURN` alternativo;
- comando `PAUSE`;
- especificadores `FORMAT` atribuídos;
- descritor de edição `H`.

1.2.4 Recursos removidos do Fortran 90

Uma vez que Fortran 90 contém o Fortran 77 como subconjunto, esta lista permaneceu vazia para o Fortran 90.

1.3 UMA REVISÃO MENOR: FORTRAN 95

Seguindo a publicação do padrão Fortran 90 em 1991, dois significativos desenvolvimentos posteriores referentes à linguagem Fortran ocorreram. O primeiro foi a continuidade na operação dos dois comitês de regulamentação do padrão da linguagem: o X3J3 e o WG5; o segundo desenvolvimento foi a criação do Fórum Fortran de Alta Performance (*High Performance Fortran Forum,* ou HPFF).

Logo no inicio de suas deliberações, os comitês concordaram na estratégia de definir uma revisão menor no Fortran 90 para meados da década de 90, seguida por uma revisão de maior escala para o início dos anos 2000. Esta revisão menor passou a ser denominada Fortran 95.

O HPFF teve como objetivo a definição de um conjunto de extensões ao Fortran tais que permitissem a construção de códigos portáveis quando se fizesse uso de computadores paralelos para a solução de problemas envolvendo grandes conjuntos de dados que podem ser representados por matrizes regulares. Esta versão do Fortran ficou conhecida como o Fortran de Alta Performance (*High Performance Fortran*, ou HPF), tendo como linguagem de base o Fortran 90, não o Fortran 77. A versão final do HPF consiste em um conjunto de instruções que contém o Fortran 90 como subconjunto. As principais extensões estão na forma de diretivas, que são vistas pelo Fortran 90 como comentários, mas que são reconhecidas por um compilador HPF. Contudo, tornou-se necessária também a inclusão de elementos adicionais na sintaxe, uma vez que nem todos os recursos desejados puderam ser acomodados simplesmente na forma de diretivas.

À medida que os comitês X3J3 e WG5 trabalhavam, este comunicavam-se regularmente com o HPFF. Era evidente que, para evitar o surgimento de dialetos divergentes de Fortran, havia a necessidade de incorporar a sintaxe nova desenvolvida pelo HPFF no novo padrão da linguagem. De fato, os recursos do HPF constituem as novidades mais importantes do Fortran 95. As outras mudanças consistem em correções, clarificações e interpretações do novo padrão. Estas se tornaram prementes quando os novos compiladores de Fortran 90 foram lançados no mercado e utilizados; notou-se uma série de erros e detalhes obscuros que demandavam reparações. Todas estas mudanças foram incluídas no novo padrão Fortran 95, que teve a sua versão inicial lançada no próprio ano de 1995.

O Fortran 95 é compatível com o Fortran 90, exceto por uma pequena alteração na função intrínseca `SIGN` (seção 8.4.2) e a eliminação de recursos típicos do Fortran 77, declarados em obsolescência no Fortran 90. Os detalhes do Fortran 95 foram finalizados em novembro de 1995 e o novo padrão ISO foi finalmente publicado em outubro de 1996. O texto completo do padrão Fortran 95 pode ser obtido em https://wg5-fortran.org/N1151-N1200/N1176.pdf.

1.3.1 RECURSOS NOVOS DO FORTRAN 95

Em relação ao Fortran 90, foram introduzidos os seguintes recursos:

- Concordância aprimorada com o padrão de aritmética de ponto flutuante binária da IEEE (IEEE 754 ou IEC 559-1989).
- Rotinas *(procedures)* puras (seção 9.2.17).
- Rotinas *(procedures)* elementais (seção 9.2.18).
- Dealocação automática de matrizes alocáveis em rotinas.
- Comando e construto `FORALL`.
- Extensões do construto `WHERE` (página 78).
- Funções especificadoras.
- Inicialização de ponteiro e a função `NULL` (seção 8.16).
- Inicialização de componentes de tipo derivado (página 38).
- Funções elementares `CEILING`, `FLOOR` e `SIGN` (seções 8.4.1 e 8.4.2).
- Funções transformacionais (8.15).

- Subrotina intrínseca `CPU_TIME` (seção 8.17.2).
- Comentário na entrada de uma lista `NAMELIST` (página 176).
- Descritores de edição com largura de campo mínimo.
- Especificação genérica na cláusula `END INTERFACE`.

1.3.2 Recursos em obsolescência no Fortran 95

Os recursos abaixo entraram na lista em obsolescência no Fortran 95 e, portanto, pouco ou nada foram comentados ao longo desta Apostila.

- Formato fixo de fonte.
- Comando `GO TO` computado.
- Declaração de variável de caractere na forma `CHARACTER*`.
- Declarações `DATA` entre comandos executáveis.
- Funções definidas em uma linha *(statement functions)*.
- Extensão assumida de caracteres quando estas são resultados de funções.

1.3.3 Recursos removidos do Fortran 95

Cinco recursos foram removidos do padrão da linguagem Fortran 95 e, portanto, não serão mais aceitos por compiladores que respeitam o padrão Fortran 95.

- Índices de laços `DO` do tipo real (qualquer espécie).
- Declaração `ASSIGN` e comando `GO TO` atribuído e uso de um inteiro atribuído por `ASSIGN` em uma declaração `FORMAT`.
- Desvio para uma declaração `END IF` a partir de um ponto fora do bloco.
- Comando `PAUSE`.
- Descritor de edição `H`.

1.4 O Fortran no Século XXI: Fortran 2003

O Fortran 2003 é novamente uma revisão grande do padrão anterior: Fortran 95. As grandes novidades introduzidas foram: um direcionamento ainda maior para programação orientada a objeto, a qual oferece uma maneira efetiva de separar a programação de um código grande e complexo em tarefas independentes e que permite a construção de novo código baseado em rotinas já existentes e uma capacidade expandida de interface com a linguagem C, necessária para que os programadores em Fortran possam acessar rotinas escritas em C/C++ e para que programadores de C possam acessar rotinas escritas em Fortran.

O padrão completo foi publicado em novembro de 2004. O texto oficial somente é disponibilizado pela ISO mediante pagamento, mas uma cópia é disponibilizada livremente pelo WG5 em https://wg5-fortran.org/N1601-N1650/N1601.pdf.[4] Uma breve descrição dos novos recursos pode ser vista em https://wg5-fortran.org/N1601-N1650/N1648.pdf.

[4] Mais informações podem ser obtidas em https://wg5-fortran.org/f2003.html.

1.4.1 RECURSOS NOVOS DO FORTRAN 2003

Os principais recursos introduzidos pelo padrão são:

- Aprimoramentos dos tipos derivados: tipos derivados parametrizados, controle melhorado de acessibilidade, construtores de estrutura aperfeiçoados e finalizadores.
- Suporte para programação orientada a objeto: extensão de tipo e herança *(inheritance)*, polimorfismo *(polymorphism)*, alocação dinâmica de tipo e rotinas *(procedures)* ligadas a tipo.
- Aperfeiçoamentos na manipulação de dados: componentes alocáveis de estruturas, argumentos mudos alocáveis, parâmetros de tipo deferidos *(deferred type parameters)*, atributo `VOLATILE`, especificação explícita de tipo em construtores de matrizes e declarações de alocação, aperfeiçoamentos em ponteiros, expressões de inicialização estendidas e rotinas intrínsecas aperfeiçoadas.
- Aperfeiçoamentos em operações de Entrada/Saída (E/S) de dados: transferência assíncrona, acesso de fluxo *(stream access)*, operações de transferência especificadas pelo usuário para tipos derivados, controle especificado pelo usuário para o arredondamento em declarações `FORMAT`, constantes nomeadas para unidades pré-conectadas, o comando `FLUSH`, regularização de palavras-chave e acesso a mensagens de erro.
- Ponteiros de rotinas *(procedure pointers).*
- Suporte para as exceções do padrão de aritmética binária de ponto flutuante ISO/IEC 559, anteriormente conhecido como padrão IEEE 754.
- Interoperabilidade com a linguagem de programação C.
- Suporte aperfeiçoado para internacionalização: acesso ao conjunto de caracteres de 4 bytes ISO 10646 and escolha de vírgula ou ponto como separador de parte inteira e parte decimal em operações de E/S numéricas formatadas.
- Integração aperfeiçoada com o sistema operacional hospedeiro: acesso a argumentos de linha de comando, variáveis de ambiente e mensagens de erro do processador.

1.4.2 RECURSOS EM OBSOLESCÊNCIA NO FORTRAN 2003

Nesta lista permanecem itens que já estavam na lista de obsolescência no Fortran 95.[5] Os critérios para inclusão nesta lista continuam sendo: recursos redundantes e para os quais métodos melhores estavam disponíveis no Fortran 95. A lista adicional dos itens em obsolescência no Fortran 2003 é:

- `IF` aritmético.
- `END DO` compartilhado por dois ou mais laços e término de um laço em uma cláusula distinta de `END DO` ou `CONTINUE`.
- `RETURN` alternativo.

1.4.3 RECURSOS REMOVIDOS DO FORTRAN 2003

Um recurso é removido do novo padrão se este for considerado redundante e praticamente sem uso pela comunidade, devido a novos recursos muito mais úteis na construção do código. A lista de recursos removidos do Fortran 2003 repete os recursos removidos do Fortran 95 (seção 1.3.3). Portanto, nenhum outro recurso foi removido.

[5] Ver seção 1.3.2.

1.4.4 Status de suporte ao padrão Fortran 2003

Embora o Fortran 2003 tenha se tornado o novo padrão da linguagem após a sua publicação, até o presente momento nem todos os compiladores suportam completamente este padrão. Uma relação dos recursos do Fortran 2003 suportados por diferentes compiladores, tanto livres (ou gratuitos), quanto proprietários, pode ser obtida em http://fortranwiki.org/fortran/show/Fortran+2003+status ou em *Chivers & Sleightholme (2019)* [3].[6]

1.5 O padrão Fortran 2008

Embora nem todos os compiladores já suportem completamente o padrão estabelecido pelo Fortran 2003, os comitês X3J3/WG5 decidiram publicar, em outubro de 2010, um novo padrão da linguagem: o Fortran 2008. Uma versão gratuita do texto final do novo padrão pode ser obtida em http://j3-fortran.org/doc/year/10/10-007r1.pdf.[7]

1.5.1 Recursos novos do Fortran 2008

O Fortran 2008, embora pelo procedimento adotado pelos comitês J3 (antes X3J3) e WG5 devesse ser uma revisão menor do Fortran 2003, apresenta algumas novidades substanciais. Alguns dos novos recursos disponíveis na linguagem são:

- Submódulos.
- Comatrizes (*Coarrays*).
- Construto `DO CONCURRENT`.
- Atributo `CONTIGUOUS`.
- Construto `BLOCK`.
- Instrução `EXIT` em (quase) qualquer construto.
- Aperfeiçoamento na instrução `STOP`.
- Rotinas internas podem ser passadas como argumentos reais.
- Rotinas de ponteiro podem apontar para rotinas internas.
- Posto máximo de matrizes aumentado para 15.
- Palavra-chave NEWUNIT= na declaração OPEN.
- Descritor de edição `G0`.
- Lista ilimitada de itens em `FORMAT`.

As seguintes mudanças foram realizadas em rotinas intrínsecas:

- `acos`, `asin`, `atan`, `cosh`, `sinh`, `tan`, e `tanh` agora aceitam argumentos complexos.
- `atan2` pode ser agora chamada simplesmente como `atan`.
- `lge`, `lgt`, `lle`, e `llt` agora aceitam argumentos da espécie ASCII.
- `maxloc` e `minloc` possuem agora um argumento opcional `back=`.
- `selected_real_kind` possui agora um argumento `radix=`.

Novas rotinas intrínsecas:

- Funções especiais:

[6] É importante ressaltar que mesmo estas listas podem se tornar rapidamente desatualizadas. O status mais recente para cada compilador deve ser fornecido pelas respectivas páginas de suporte.

[7] Mais informações podem ser obtidas em https://wg5-fortran.org/f2008.html.

  - funções trigonométricas hiperbólicas inversas: `acosh`, `asinh`, e `atanh`.
  - Funções de Bessel de argumento real: `bessel_j0`, `bessel_j1`, `bessel_jn`, `bessel_y0`, `bessel_y1` e `bessel_yn`.
  - Função erro: `erf`, `erfc` e `erfc_scaled`.
  - Função gama: `gamma` e `log_gamma`.
  - Função distância Euclideana: `hypot` e norma L2: `norm2`.

- Operações bit-a-bit (*bitwise operations*):
  - Bit sequence comparisons: `bge`, `bgt`, `ble` e `blt`.
  - Combined shifting: `dshiftl` e `dshiftr`.
  - Counting bits: `leadz`, `trailz`, `popcnt` e `poppar`.
  - Masking bits: `maskl` e `maskr`.
  - Shifting bits: `shifta`, `shiftl` e `shiftr`.
  - Merging bits: `merge_bits`.
  - Bit transformational functions: `iall`, `iany` e `iparity`.

- Coarray intrinsics:
  - Convert a cosubscript to an image index: `image_index`.
  - Cobounds of a coarray: `lcobound` e `ucobound`.
  - Number of images: `num_images`.
  - Image index or cosubscripts: `this_image`.

- Outros:
  - Test for the contiguous attribute: `is_contiguous`.
  - Size of an element in bits: `storage_size`.
  - Tests for the number of true values being odd: `parity`.
  - Search for a value in an array: `findloc`.
  - Shell commands: `execute_command_line`.
  - Define and reference variables atomically: `atomic_define` e `atomic_ref`.

Adições a módulos intrínsecos:

- `ISO_FORTRAN_ENV`:
  - Information about the compiler: `compiler_version` e `compiler_options`.
  - Named constants for selecting kind values.

- `IEEE_ARITHMETIC`:
  - `ieee_selected_real_kind` now has a `radix=` argument.

- `ISO_C_BINDING`:
  - `c_sizeof` returns the size of an array element in bytes.

1.5.2 RECURSOS EM OBSOLESCÊNCIA NO FORTRAN 2008

A lista dos recursos obsolescentes no Fortran 2008 contém itens que já estavam presentes nas listas do Fortran 95 (seção 1.3.2) e do Fortran 2003 (seção 1.4.2). Adicionalmente, foi incluído:

- Instrução `ENTRY`.

1.5.3 Recursos removidos do Fortran 2008

Em adição aos recursos já removidos pelo Fortran 95 (seção 1.3.3), os seguintes recursos foram também excluídos do padrão:

- Formato de controle vertical.

1.5.4 Status do suporte ao Fortran 2008

Uma relação dos recursos do Fortran 2008 suportados por diferentes compiladores, tanto livres (ou gratuitos), quanto proprietários, pode ser obtida em http://fortranwiki.org/fortran/show/Fortran+2008+status ou em *Chivers & Sleightholme (2019)* [3].[8]

1.6 O novo padrão: Fortran 2018

O Fortran 2018 é uma extensão pequena ao Fortran 2008. Os principais detalhes já haviam sido definidos pelos comitês em 2015, mas o novo padrão foi lançado somente em novembro de 2018.[9] O texto completo do novo padrão pode ser obtido em https://j3-fortran.org/doc/year/18/18-007r1.pdf e uma descrição sucinta pode ser vista em The New Features of Fortran 2018, por John Reid.[10]

1.6.1 Recursos novos do Fortran 2018

O Fortran 2018 estende o Fortran 2008 em dois recursos básicos:

Interoperabilidade adicional entre as linguagens Fortran e C, com a introdução das estruturas denominadas `C descriptors`, as quais permitem que sejam passados os endereços de argumentos que não eram interoperáveis, ao invés dos objetos propriamente. O acesso aos objetos passados desta maneira às funções do C é possibilitado também pela definição do arquivo de cabeçalhos (*header file*) `ISO_Fortran_binding.h` [4].

Recursos adicionais de paralelismo no Fortran, com a definição de *Times* (*Teams*) dentro dos coarrays.

Conformidade com as normas ISO/IEC/IEEE 60559:2011, as quais definem o novo padrão para a artimética de ponto flutuante, adotado pela maioria dos fabricantes de hardware e desenvolvedores de software. Os novos recursos são adicionados aos módulos intrínsecos `IEEE_ARITHMETIC`, `IEEE_EXCEPTIONS` e `IEEE_FEATURES`. Maiores informações acerca da norma IEEE 60559:2011 podem ser obtidas em https://www.iso.org/standard/57469.html.

1.6.2 Recursos em obsolescência no Fortran 2018

A lista repete recursos já tornados obsoletos nas listas do Fortran 95 (seção 1.3.2), Fortran 2003 (seção 1.4.2) e Fortran 2008 (seção 1.5.2) e inclui:

- Declaração `DO` com rótulo.
- Declarações `COMMON` e `EQUIVALENCE` e a unidade de programa `BLOCK DATA`.
- Nomes específicos para funções intrínsecas.
- Declaração e construto `FORALL` (criados pelo Fortran 95).[11]

[8] Ver observação na nota de rodapé 6.
[9] Ver anúncio em Fortran 2018 has been published!.
[10] Mais informações podem ser obtidas em https://wg5-fortran.org/f2018.html.
[11] As descrições destes recursos podem ser visualizadas em Introdução ao Fortran 90/95, disponível em http://professor.ufrgs.br/rgaelzer/pages/disciplinas-ufpel.

1.6.3 RECURSOS REMOVIDOS DO FORTRAN 2018

Em adição aos recursos já removidos pelo Fortran 95 (seção 1.3.3) e pelo Fortran 2008 (seção 1.5.3), também foram eliminados do padrão:

- Comando `IF` aritmético.
- Construto `DO` que não está na forma de um bloco.

1.6.4 STATUS DO SUPORTE AO FORTRAN 2018

Devido ao seu recente lançamento, os novos recursos criados pelo Fortran 2018 são suportados (parcialmente) por poucos compiladores. Uma lista do suporte pode ser obtida em http://fortranwiki.org/fortran/show/Compiler+Support+for+Modern+Fortran.

1.7 RECURSOS ADICIONAIS AO FORTRAN MODERNO

Ao longo de seus mais de cinquenta anos, o Fortran acumulou uma quantidade impressionante de códigos e bibliotecas, muitos dos quais são acessíveis. Frisando que uma listagem completa de todos os recursos é impossível, pode-se mencionar os seguintes:

Listas de discussão. Salas virtuais de diálogos, onde é possível buscar a solução de dúvidas ou formular perguntas, as quais serão (tentarão) respondidas por outros programadores com os mais variados graus de conhecimento e experiência. Alguns dos assíduos participantes destas listas são autores de livros sobre Fortran e/ou desenvolvedores de compiladores ou bibliotecas. As principais listas de discussão são:

- Lista sobre Fortran no Google Groups.
- Listas do compilador Intel® Fortran para Linux e macOS e para Windows.

Bibliotecas e plataformas de desenvolvimento. Onde são acessíveis recursos tanto oriundos de décadas de experiência no desenvolvimento de métodos numéricos quanto de novos projetos desenvolvidos na linguagem. Pode-se destacar:

- Biblioteca NetLib em https://www.netlib.org.
- Biblioteca SLATEC em https://people.sc.fsu.edu/~jburkardt/f_src/slatec/slatec.html.
- Biblioteca fgsl: interface em Fortran para GSL (GNU Scientific Library): https://github.com/reinh-bader/fgsl.
- Plataforma de desenvolvimento GitHub. Projetos em Fortran: https://github.com/topics/fortran.
- Plataforma de desenvolvimento SourceForge. Projetos em Fortran: https://sourceforge.net/directory/os:linux/?q=fortran.

Compiladores. Compiladores de Fortran:

(Parcialmente) gratuitos:

- Gfortran: https://gcc.gnu.org/wiki/GFortran.
- Intel® Fortran (https://software.intel.com/en-us/fortran-compilers). Versão livre por 30 dias. Estudantes, professores ou desenvolvedores de software podem adquirir uma licença gratuita. Ver detalhes em https://software.intel.com/en-us/parallel-studio-xe/choose-download.
- PGI (Community Edition): https://www.pgroup.com/products/community.htm.

Comerciais:

- Absoft: https://www.absoft.com.
- NAG Fortran: https://www.nag.co.uk/nag-compiler.

O uso de um compilador para testar os exemplos contidos nesta Apostila é imprescindível para o aprendizagem de qualquer linguagem de programação.

Uma lista mais longa e detalhada de recursos pode ser obtida a partir do texto Fortran Resources, de Ian D. Chivers & Jane Sleghtholme, disponível em https://www.fortranplus.co.uk/app/download/30202494/fortran_resources.pdf.

1.8 Comentários sobre a bibliografia

Para escrever esta apostila, fiz uso de um número restrito de publicações, algumas das quais são de livre acesso através da internet.

- A principal fonte de informação sobre o Fortran Moderno foi o amplo livro de Michael Metcalf, John Reid e Malcolm Cohen: *Modern Fortran Explained. Incorporating Fortran 2018* [1].
- Outros livros relevantes:
  - Stephan J. Chapman. *Fortran for Scientists and Engineers* (4th. Ed.) [2].
  - Norman S. Clerman & Walter Spector. *Modern Fortran. Style and Usage* [5].
- O padrão do Fortran 2018, disponível em https://j3-fortran.org/doc/year/18/18-007r1.pdf, também foi consultado.
- O manual de referência à Linguagem Fortran que acompanha o compilador Intel também foi consultado [6].

1.9 Observações sobre a apostila e agradecimentos

Esta Apostila consiste em uma nova versão do texto Introdução ao Fortran 90/95, o qual foi empregado para ministrar disciplina de métodos computacionais aplicados à Física na Universidade Federal de Pelotas até 2012 e que ainda está acessível em http://professor.ufrgs.br/rgaelzer/pages/disciplinas-ufpel. A presente versão (Introdução ao Fortran Moderno) não faz menção a um padrão em particular da linguagem, abordando outrossim recursos suportados tanto pelo padrão mais recente quanto pelos compiladores nos quais os códigos foram testados.

Em particular, este texto não irá abordar recursos do Fortran 77 ou outros padrões posteriores que sejam hoje considerados obsolescentes. Programadores em Fortran 77 podem ainda consultar o texto Introdução ao Fortran 90/95 ou outros textos tais como: Fortran 90. A Conversion Course for Fortran 77 Programmers, Fortran90 for Fortran77 Programmers ou Conversion from Fortran 77 to Modern Fortran. Uma excelente revisão de Fortran 77 é também apresentada por Clive Page em Professional Programmer's Guide to Fortran 77.

Esta Apostila propõe-se a apresentar uma discussão abrangente de uma parte dos recursos oferecidos pelo padrão atual da linguagem. Contudo, as quatro grandes áreas a seguir não serão abordadas:

- Programação orientada a objetos.
- Coarrays.
- Interoperabilidade com C.
- Controle de exceções de ponto flutuante e conformidade com o padrão IEEE 754-2008.

Embora sejam importantes, estes recursos estão além do nível de uma introdução à linguagem.

Todos os códigos ou partes de códigos apresentados neste texto estão de acordo com o padrão mais recente da linguagem e, portanto, devem ser aceitos por compiladores (também nas versões mais recentes) em qualquer arquitetura de hardware. Contudo, como foi ressaltado ao longo das seções 1.2 – 1.6, embora seja razoável esperar que todos os compiladores atuais suportem pelo menos o padrão Fortran 95, o nível de suporte a recursos criados a partir do Fortran 2003 é bastante variado. Os recursos apresentados nesta Apostila foram testados e são aceitos nos seguintes compiladores instalados no sistema operacional Linux:

Intel® Fortran Compiler. Versão empregada:

```
user@machine|dir>ifort -V
Intel(R) Fortran Intel(R) 64 Compiler for applications running on
Intel(R) 64,
Version 19.0.1.144 Build 20181018
Copyright (C) 1985-2018 Intel Corporation.  All rights reserved.
```

GFortran. Versão empregada:

```
user@machine|dir>gfortran --version
GNU Fortran (GCC) 9.1.1 20190503 (Red Hat 9.1.1-1)
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is
    NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE.
```

Eventualmente, poderá ocorrer que um determinado recurso é suportado por somente um dos compiladores nas versões acima. Neste caso, será impressa uma imagem na margem lateral indicando qual compilador ainda não suporta o recurso em questão. As imagens são: para o compilador Intel® Fortran e para o GFortran.

Por se tratar de uma obra em constante revisão, esta apostila pode conter (e conterá, invariavelmente) uma série de erros de ortografia, pontuação, acentuação, *etc*. Certos pontos poderiam também ser melhor discutidos e podem conter até informações não completamente corretas.

Durante o desenvolvimento e divulgação desta apostila, algumas pessoas contribuiram com o seu aprimoramento ao apontar erros e inconsistências e ao oferecer sugestões quanto a sua estrutura e conteúdo. Qualquer contribuição é bem vinda e pode ser enviada ao endereço eletrônico: rudi.gaelzer@ufrgs.br.

Gostaria de agradecer publicamente as contribuições das seguintes pessoas: Leandro Tezani, Antonio Barbosa.