

Métodos Computacionais em Física

Aula 02

Introdução à Linguagem FORTRAN

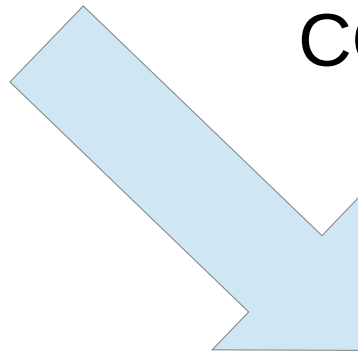
Criando um programa

- O programa começa com um objetivo;
- Ele é composto de uma série de ordens escritas em uma linguagem entendida pelo usuário;
- Essas ordens devem ser traduzidas em uma linguagem entendida pelo computador;
- Isto é feito por um compilador, que cria um arquivo executável.

```
Laksdjfjksdf;llaskdfí  
Safjnasflksflkjsf;ksf  
Sdfgsdfgsdgsdgsdf  
g  
Dsfgsdfgdsfgsdfgsd  
fg  
Dfgsdffgdfs  
Gdfgsdffgsdfgsdfgd  
fs  
gdgsdfgdfsgdfsg
```

Program.f90

COMPILADOR



```
11001100001  
00111000111  
00110011110  
11000000011
```

Program.exe

- FORTRAN 77
- FORTRAN 90
- FORTRAN 95
- FORTRAN 2003
- FORTRAN 2008

Como escrever um programa?

- Entenda o problema claramente;
- Analise o problema e o quebre em pedaços fundamentais;
- Escreva um código com o plano desenvolvido no item anterior;
- Teste e repita os dois passos anteriores até que o programa funcione em todos os casos que vc desejar.

Estrutura básica de um programa em FORTRAN

Hello world!!!

Soma de dois números.

Compilando e executando um programa

- Pelo terminal, entre na pasta onde o arquivo .f90 está;
- Digite:
`gfortran -o exec.x program.f90`
- Isto criará o arquivo exec.x;
- Execute o programa digitando:
`./exec.x`

Formato

- Espaços e vírgulas podem separar variáveis;
- Comentários começam com uma exclamação !
- Uma linha pode conter até 132 caracteres;
- Uma linha pode ter mais de um statement (separando-os por ponto e vírgula);
- Um & no fim de uma linha indica que a próxima linha é uma continuação dela (começe a linha seguinte com outro &)
- Um comando pode ter até 39 linhas de continuação;

Statement labels

- Pode-se usar os cinco primeiros caracteres para escrever um inteiro que rotule a linha;
- Em FORTRAN 77, usa-se o sexto caractere para definir se a linha é de continuação ou não.
-
- Particularidades do formato 77: livro

Parte II - Variáveis

Tipos Fundamentais

- Inteiros;
- Reais;
- Caracteres;
- Complexos;
- Lógicos.

Inteiros

```
PROGRAM integers
```

```
IMPLICIT NONE
```

```
INTEGER :: i, j, soma
```

```
i=10
```

```
j=20
```

```
soma=i+j
```

```
PRINT*, soma
```

```
END PROGRAM integers
```

10

i

integer

20

j

integer

30

soma

integer

IMPLICIT NONE

- Declaração implícita:
 - I – N: inteiros
 - outros casos: real
- Implicit None requer declaração de todas as variáveis.

Reais

```
PROGRAM reais
```

```
IMPLICIT NONE
```

```
REAL:: a,b,soma
```

```
a=2.7
```

```
b=3.1
```

```
soma=a+b
```

```
PRINT*,soma
```

```
END PROGRAM reais
```

2.7

a

real

3.1

b

real

≈ 5.8

soma

real

Expressões numéricas e atribuições

$$a = b + c$$

$$a = a + b$$

$$\begin{array}{ccccc} a & = & b & + & c \\ \text{integer} & & \text{integer} & & \text{integer} \end{array}$$

$$\begin{array}{ccccc} a & = & b & + & c \\ \textit{real} & & \textit{real} & & \textit{real} \end{array}$$

$$\begin{array}{ccccc} a & = & b & + & c \\ \text{real} & & \text{integer} & & \text{integer} \end{array}$$

Converte-se o resultado para real

$$\begin{array}{ccccc} a & = & b & + & c \\ \text{integer} & & \text{real} & & \text{real} \end{array}$$

truncamento

$$\begin{array}{ccccc} a & = & b & + & c \\ \text{integer} & & \text{real} & & \text{integer} \end{array}$$

Converte-se c para real

Mais expressões numéricas

$$a = b + c$$

$$a = b - c$$

Mais expressões numéricas

$$a = b * c$$

$$a = b / c$$

$$a = b ** c$$

Operador	Prioridade
**	alta
* ou /	média
+ ou -	baixa

$a = b + c * d / e - f ** g / h + i * j + k$

$a = b * c / d$

$a = c / d * b$

$$a=(b/c)*d$$

$$a=b/(c*d)$$

a – real

b=100.0 (real)

c=9 (inteiro)

d=10 (inteiro)

a=b*c/d= 90.0

a – real

b=100.0 (real)

c=9 (inteiro)

d=10 (inteiro)

a=c/d*b= 0.0

Adição comutativa?

- Suponha precisão de 4 dígitos
- $A=5.549$
- $B=6.874$
- $C=-3.432$
- $D=A+B+C=12.423+C=8.988$
- $D=A+C+B=3.442+B=8.991$

Lendo e imprimindo dados

- READ*, var_1, var_2, var_3
- PRINT*, var1, var2, var3

```
PROGRAM read_print
```

```
IMPLICIT none
```

```
REAL:: a,b,c,d,e
```

```
PRINT*, 'Give me three numbers'
```

```
READ*, a, b, c
```

```
d=a+b
```

```
e=a*b*c
```

```
PRINT*, d, e
```

```
END PROGRAM read_print
```

Imprimindo caracteres

- Coloque “texto”
- Ou coloque 'texto'

Variáveis Caracteres

- Declara-se como:

`CHARACTER(LEN=4):: variable`

- ou

`CHARACTER(4):: variable_a, variable_b`

- ou

`CHARACTER*4:: variable_a, variable_b`

- Ou

`CHARACTER:: variable_a*4, variable_b*2`

Variáveis Caracteres

- Mais informações em uma aula posterior...

Variáveis Lógicas

- Declara-se como
LOGICAL:: var_a
- Onde var_a pode assumir dois valores:
 - var_a=.true.
 - var_a=.false.