

Introdução ao Fortran 90 - 5

Alexandre Diehl

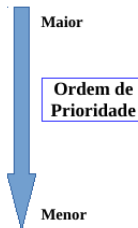
Departamento de Física – UFPel

Estrutura de Repetição DO

Definição

A estrutura de **repetição DO** fornece um mecanismo para controle de desvio de fluxo, que permite que um **bloco de comandos** seja **executado** de forma **repetida**, através de um número **finito** ou **infinito** de **repetições**.

Tipo	Operador						Associatividade
Aritmético	**						Direita para Esquerda
	*			/			Esquerda para Direita
	+			-			Esquerda para Direita
Relacional	<	<=	>	>=	==	/=	Nenhuma
Lógico	.NOT.						Direita para Esquerda
	.AND.						Esquerda para Direita
	.OR.						Esquerda para Direita
	.EQV.			.NEQV.			Esquerda para Direita



Estrutura de Repetição DO

Caso 1: Comando DO ... END DO finito

→ Permite que sejam feitas um número finito de iterações, onde a cada iteração os comandos listados dentro do bloco serão executados.

DO <controle> = <início>, <fim>, [, <passo>]

<comando 1>

...

<comando n>

END DO

- 1 A variável <controle>, do tipo **INTEGER**, é inicializada com o valor da expressão <início>:
 - se <controle> for um valor contido entre <início> e <fim>, os comandos listados dentro da estrutura serão executados.
 - caso contrário, o próximo comando imediatamente posterior ao **END DO** será executado.
- 2 Após a execução dos comandos, o <passo> é adicionado à variável <controle> e o fluxo de execução é desviado para o item 1 acima, com os testes listados feitos novamente.

Caso 1: Comando DO ... END DO finito

→ Permite que sejam feitas um **número finito de iterações**, onde a cada iteração os comandos listados dentro do bloco serão executados.

DO <controle> = <início>, <fim>

<comando 1>

...

<comando n>

END DO

- Se o <passo> é **omitido**, será **adicionado o valor 1** à variável <controle>:
 - neste caso, <início> deve ser menor do que <fim>;
 - <início> e <fim> devem ser variáveis ou expressões do tipo **INTEGER**.

Caso 1: Comando DO ... END DO finito

→ Permite que sejam feitas um **número finito de iterações**, onde a cada iteração os comandos listados dentro do bloco serão executados.

DO <controle> = <início>, <fim>, <passo>

<comando 1>

...

<comando n>

END DO

- Se o <passo> é maior do que zero, será adicionado o valor do <passo> à variável <controle>:
 - neste caso <início> deve ser menor do que <fim>
 - <início> e <fim> devem ser variáveis ou expressões do tipo **INTEGER**.

Caso 1: Comando DO ... END DO finito

→ Permite que sejam feitas um **número finito de iterações**, onde a cada iteração os comandos listados dentro do bloco serão executados.

DO <controle> = <início>, <fim>, <passo>

<comando 1>

...

<comando n>

END DO

- Se o <passo> é **menor do que zero**, será **adicionado** o valor do <passo> à variável <controle>:
 - neste caso <início> deve ser **maior** do que <fim>
 - <início> e <fim> devem ser variáveis ou expressões do tipo **INTEGER**.

Caso 1: Comando DO ... END DO finito

→ Permite que sejam feitas um número finito de iterações, onde a cada iteração os comandos listados dentro do bloco serão executados.

```
DO <controle> = <início>, <fim>, <passo>  
  <comando 1>  
  ...  
  <comando n>  
END DO
```

- O <passo> não pode ser igual a zero.
- Nunca mude o valor das variáveis <controle>, <início>, <fim> e <passo> dentro da estrutura de repetição.
- No Fortran 77 é permitido o uso de uma variável <controle> do tipo REAL. No Fortran 90 não é permitido.

Estrutura de Repetição DO

Caso 1: Comando DO ... END DO finito

Exemplo: Cálculo do fatorial de um inteiro N

```
INTEGER :: i, N, fatorial
PRINT*, 'Digite o valor de N: '
READ*, N
fatorial = 1
DO i = 1, N
    fatorial = fatorial * i
END DO
print*, 'O fatorial de N= ', N, ' vale ', fatorial
END
```


Caso 2: Comando DO WHILE ... END DO finito

→ Permite que sejam feitas um número finito de iterações, enquanto uma condição lógica for atendida; a cada iteração os comandos listados dentro do bloco serão executados.

DO WHILE (<condição>)

<comando 1>

...

<comando n>

END DO

- 1 Se a <condição> for avaliada como **.TRUE.** os comandos listados dentro da estrutura são executados.
- 2 Se a <condição> for avaliada como **.FALSE.** o comando imediatamente posterior ao **END DO** será executado.

Caso 2: Comando DO WHILE ... END DO finito

Exemplo: Cálculo do fatorial de um inteiro N

```
INTEGER :: i, N, fatorial
PRINT*, 'Digite o valor de N: '
READ*, N
fatorial = 1
i = 2
DO WHILE (i <= N)
  fatorial = fatorial * i
  i = i + 1
END DO
print*, 'O fatorial de N = ', N, ' vale ', fatorial
END
```

Caso 3: Comando DO ... END DO infinito

→ Permite que sejam feitas um **número infinito de iterações**; a cada iteração os comandos listados dentro do bloco serão executados.

DO

<comando 1>

...

<comando n>

END DO

- 1 O **fluxo** de execução será **desviado** para **fora da estrutura** usando o comando **EXIT**:

→ neste caso, nenhum dos comandos entre **EXIT** e **END DO** é executado.

Caso 3: Comando DO ... END DO infinito

Exemplo: Cálculo do fatorial de um inteiro N

```
INTEGER :: i, N, fatorial
PRINT*, 'Digite o valor de N: '
READ*, N
fatorial = 1
i = 2
DO
  IF (i > N) EXIT
  fatorial = fatorial * i
  i = i + 1
END DO
print*, 'O fatorial de N= ', N, ' vale ', fatorial
END
```

Caso 3: Comando DO ... END DO infinito

→ Permite que sejam feitas um **número infinito de iterações**; a cada iteração os comandos listados dentro do bloco serão executados.

DO

<comando 1>

...

<comando n>

END DO

- ❶ O **fluxo** de execução será **desviado dentro da estrutura** usando o comando **CYCLE**:

→ neste caso, nenhum dos comandos entre **CYCLE** e **END DO** é executado e uma nova iteração dentro do laço é executada.

Estrutura de Repetição DO

Caso 3: Comando DO ... END DO infinito

Exemplo: Soma dos primeiros N inteiros ímpares

```
integer :: i, impar, N, soma
print*, 'Quantos ímpares para somar?'
read*, N

soma = 0
impar = 0
i = 0
do
  i = i + 1
  IF (impar >= N) EXIT
  IF (mod(i,2) == 0) CYCLE ! exclui inteiros pares
  impar = impar + 1
  soma = soma + i
end do
print*, 'A soma dos ', N, ' primeiros ímpares vale ', soma
end
```

TAREFA 3:

Um foguete é lançado verticalmente e sobe com uma aceleração constante de 20 m/s^2 durante um minuto. O seu combustível acaba e ele continua a mover-se como uma partícula em queda livre.

Faça um programa em **FORTRAN 90** que permita obter a posição y , a velocidade v e a aceleração a do foguete, de forma iterativa, desde o ponto de lançamento até o ponto em que o foguete volta para o solo. O programa deve ter as seguintes características:

- O programa deve conter estruturas do tipo condicional **IF** e de repetição **DO**.
- A saída deve apresentar o tempo t , a posição y , a velocidade v e a aceleração a .